

**ANALISIS PERBANDINGAN AUTOMATION API TEST
MENGGUNAKAN KATALON DAN KARATE**

MAKALAH

**Untuk Memenuhi Salah Satu Syarat Kelulusan
BRILiaN Future Leader Program (BFLP) IT
Angkatan 4**



Oleh:

REEZKY ILLMAWATI

**BRILIAN FUTURE LEADER PROGRAM (BFLP) IT ANGKATAN 4
CORPORATE UNIVERSITY
PT. BANK RAKYAT INDONESIA (PERSERO) TBK
2022**

SURAT PERNYATAAN TIDAK PLAGIAT

Sebagai peserta pendidikan BRILiaN Future Leader Program (BFLP) IT PT Bank Rakyat Indonesia (Persero) Tbk, kami yang bertanda tangan dibawah ini:

Nama : Reezky Illmawati
Program Pendidikan : BFLP IT
Angkatan : 8
Judul Makalah : *“Analisis Perbandingan Integrasi Automation API Test Menggunakan Karate dan Katalon”*

1. Makalah kami, tugas akhir dalam pendidikan BFLP ini adalah asli dan belum pernah diajukan untuk mendapatkan predikat lulus dari BFLP sebelumnya.
2. Makalah ini bukan saduran / terjemahan namun murni ide / gagasan, rumusan, dan pelaksanaan penelitian / implementasi saya, tanpa bantuan pihak lain kecuali arahan pembimbing makalah dan narasumber.
3. Dalam makalah ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Kami menyerahkan hak atas makalah ini kepada PT Bank Rakyat Indonesia (Persero) Tbk dan oleh karenanya PT. Bank Rakyat Indonesia (Persero) Tbk berhak melakukan pengelolaan atas makalah ini sesuai dengan norma hukum dan etika yang berlaku.
5. Pernyataan ini kami buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan kesalahan dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Jakarta, 25 Juni 2022



Reezky Illmawati

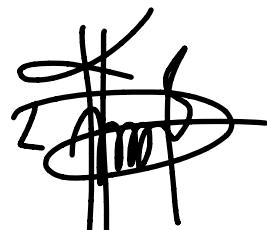
LEMBAR PERSETUJUAN PEMBIMBING MAKALAH

Yang bertanda tangan dibawah ini, menyatakan bahwa telah menyetujui makalah sebagai berikut:

Nama : Reezky Illmawati
Program : BFLP IT
Angkatan : 8
Judul Makalah : Analisis Perbandingan Integrasi Automation API
Test Menggunakan Karate dan Katalon.

Serta telah dilakukan revisi dan perbaikan sesuai hasil proses pembimbingan yang telah dilakukan.

Jakarta, 13 Juli 2022



Noviana Pramitasari

Team Leader E-Channel & Front
End Acceptance

FORMULIR BIMBINGAN MAKALAH



PT. BANK RAKYAT INDONESIA (PERSERO) Tbk.

BRI CORPORATE UNIVERSITY

FORMULIR COACHING / MENTORING

BRILIAN FUTURE LEADER PROGRAM (BFLP)

Tanggal Coaching : 13 Juni 2022

BFLP Angkatan : 8

Nama Peserta : Reezky Illmawati

Pembahasan : Menentukan Judul dari Makalah Kompre

1	Pembahasan mengenai konsep makalah kompre yang akan digunakan mengenai Automation Testing
2	Menentukan judul makalah berdasarkan konsep makalah mengenai Automation Testing
3	Membahas materi terkait konsep makalah automation testing seperti penjabaran detail tools automation testing yang akan diimplementasikan pada makalah kompre
4	Menentukan batasan masalah dari makalah kompre

Nama Coach :

A handwritten signature in black ink, appearing to read 'Noviana Pramitasari'.

Noviana Pramitasari

Team Leader E-Channel &
Front End Acceptance

- Lembar Pembahasan boleh lebih dari 1 pada setiap mentoring



PT. BANK RAKYAT INDONESIA (PERSERO) Tbk.

BRI CORPORATE UNIVERSITY

FORMULIR COACHING / MENTORING
BRILIAN FUTURE LEADER PROGRAM (BFLP)

Tanggal Coaching : 30 Juni 2022

BFLP Angkatan : 8

Nama Peserta : Reezky Illmawati

Pembahasan :	Menentukan Judul dari Makalah Kompre
---------------------	--------------------------------------

1	Revisi Materi Presentasi : -Memperbaiki tampilan -Mengoreksi susunan slide -Menambahkan konten dengan usability dari kedua tools automation
2	Revisi Makalah : -Mengoreksi tujuan dan batasan makalah -Revisi Bab 3 dan Bab 4

Nama Coach :

Noviana Pramitasari

Team Leader E-Channel &
Front End Acceptance

- Lembar Pembahasan boleh lebih dari 1 pada setiap mentoring



PT. BANK RAKYAT INDONESIA (PERSERO) Tbk.

BRI CORPORATE UNIVERSITY

FORMULIR COACHING / MENTORING
BRILIAN FUTURE LEADER PROGRAM (BFLP)

Tanggal Coaching : 13 Juli 2022

BFLP Angkatan : 8

Nama Peserta : Reezky Illmawati

Pembahasan :	Menentukan Judul dari Makalah Kompre
---------------------	--------------------------------------

1	Presentasi mengenai makalah kompre terkait analisis perbandingan tools automation menggunakan katalon studio dan karate
2	Revisi presentasi kompre : -Mengoreksi susunan slide mengenai struktur project dari kedua tools -Penjelasan lebih detail terkait Reporting hasil dari performance testing -Menambahkan kuesioner terkait usability dari kedua tools karate dan katalon
3	Revisi makalah kompre : -Menghilangkan pembahasan terkait struktur project dari Bab III dan Bab IV -Revisi terkait penjelasan dari hasil resporting performance test -Menambahkan kuesioner terkait usability dari kedua tools karate dan katalon

Nama Coach :

Noviana Pramitasari

Team Leader E-Channel &
Front End Acceptance

- Lembar Pembahasan boleh lebih dari 1 pada setiap mentoring

PRESENTASI MAKALAH

The slide features a blue header section with white text. At the top right are the logos for BFLP (Binaan Finansial Lembaga Pendidikan) and BANK BRI. The main title is "Analisis Perbandingan Automation API Test menggunakan Katalon dan Karate". Below the title is the author's name, Reezky Illmawati (00310304), and the institution, BFLP IT 8.

**Analisis Perbandingan
Automation API Test
menggunakan Katalon dan
Karate**

Reezky Illmawati (00310304)
BFLP IT 8



::: Latar Belakang

The diagram illustrates the comparison between Katalon and Karate. It features a central circular diagram divided into three segments: Web Testing (grey), API Testing (blue), and Mobile Testing (blue). Below this, there is a photograph of two people working on a large computer monitor displaying code and a gear icon. To the right, there are logos for Katalon (with text "Penggunaan CPU Thread Aktif Memory") and Karate (with a stylized figure icon).

Automation Test

Web Testing | API Testing | Mobile Testing

API Testing

Functionality Testing : menguji fungsionalitas API .

Usability Testing : menguji API dari segi user friendly

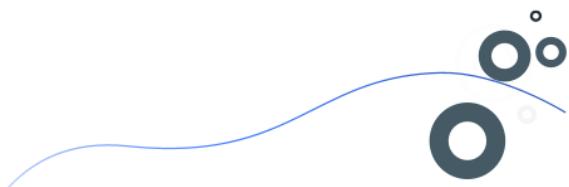
Performance Testing : Mengukur kemampuan performa suatu aplikasi sampai suatu batas tertentu

Katalon
Penggunaan CPU
Thread Aktif
Memory

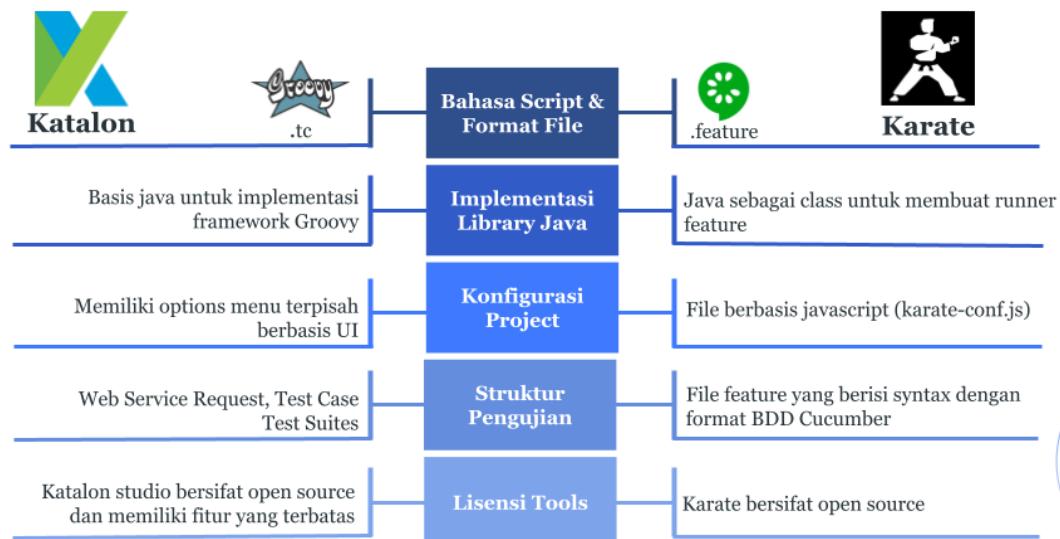
Karate

Tujuan Penelitian

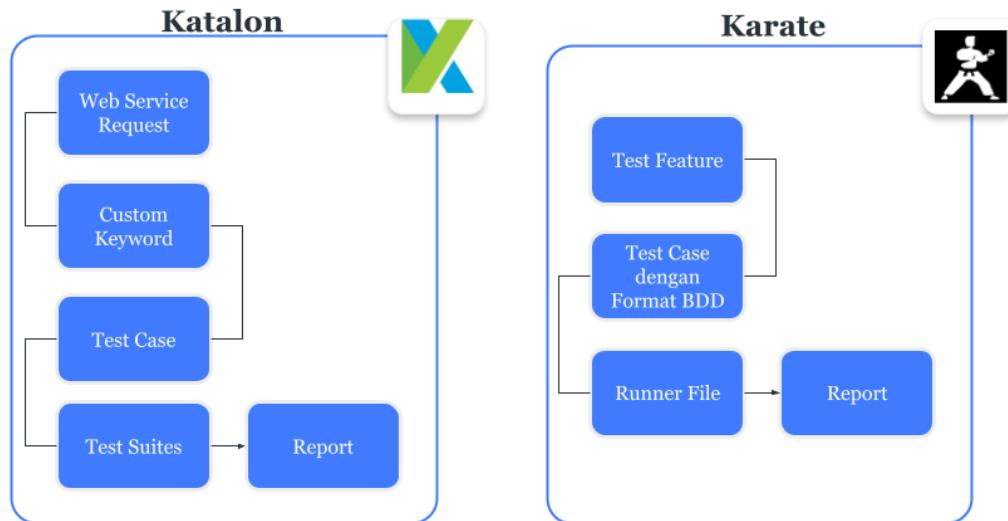
1. Analisis Hasil dari **Performa Eksekusi tools**
2. Analisis **Penggunaan Sumber Daya tools**
3. Perbandingan **Tingkat Usability Tools**
4. Analisis **Hasil Performance test API** pada tools Eksisting BRI dan Karate
5. Memberikan **rekomendasi tools automation** untuk pengujian aplikasi di BRI



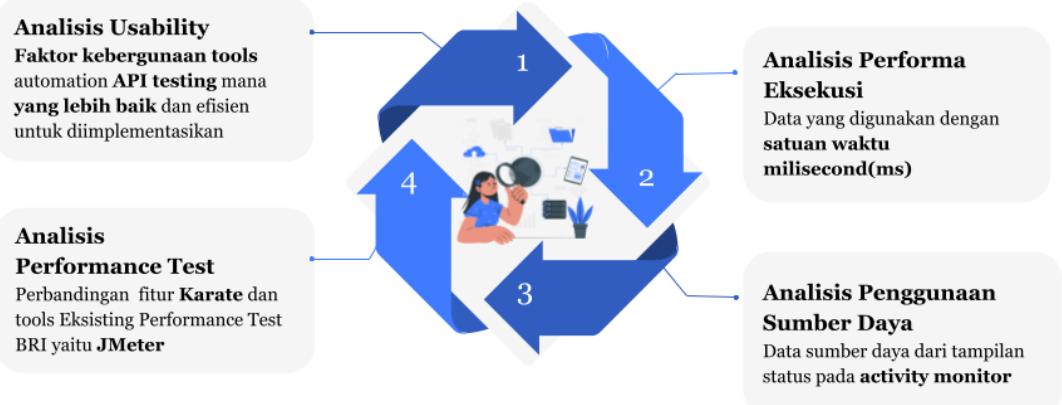
Perbandingan Tools



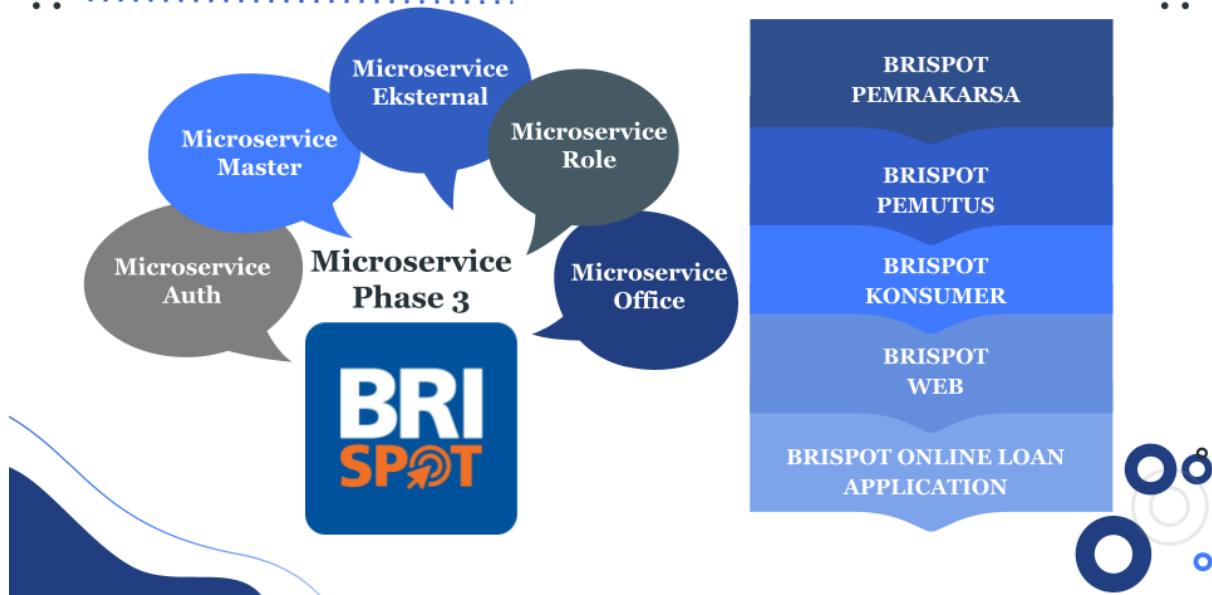
Proses Automation API Test



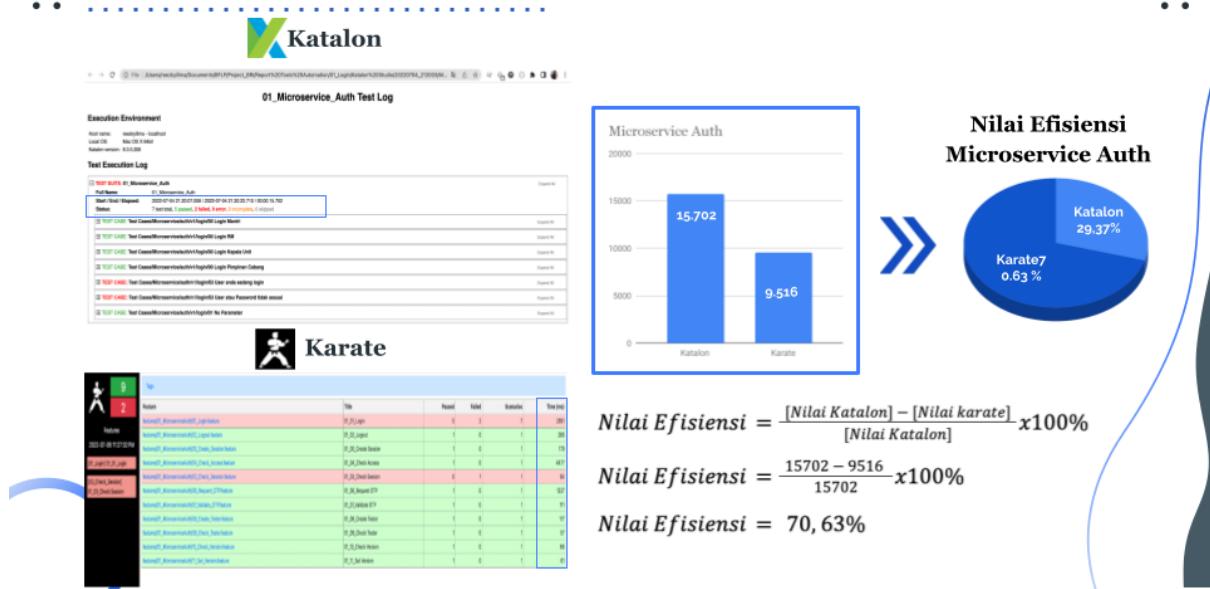
Teknik Analisis



APLIKASI BRISPOT



Performa Eksekusi



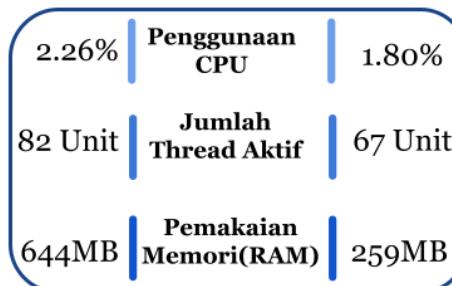
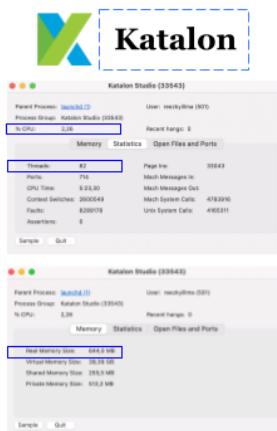
Performa Eksekusi

Alur Pengujian	Waktu Eksekusi (milisecond)		Nilai Efisiensi (%)
	Katalon	Karate	
1 Microservice Auth	15.702	9.516	70,63
2 Microservice Master	104.123	5.927	94,30
3 Microservice Eksternal	26.273	8.213	68,87
4 Microservice Role	9.638	1.862	80,68
5 Microservice Office	5.705	1.021	82,10

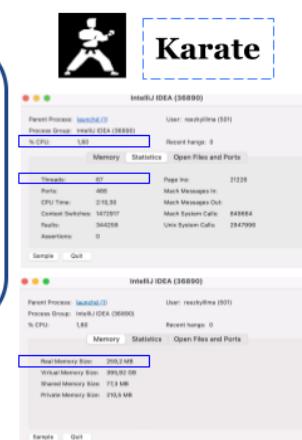


Rata-rata nilai efisiensi dari penggunaan karate dibandingkan dengan Katalon studio adalah **79,32%**

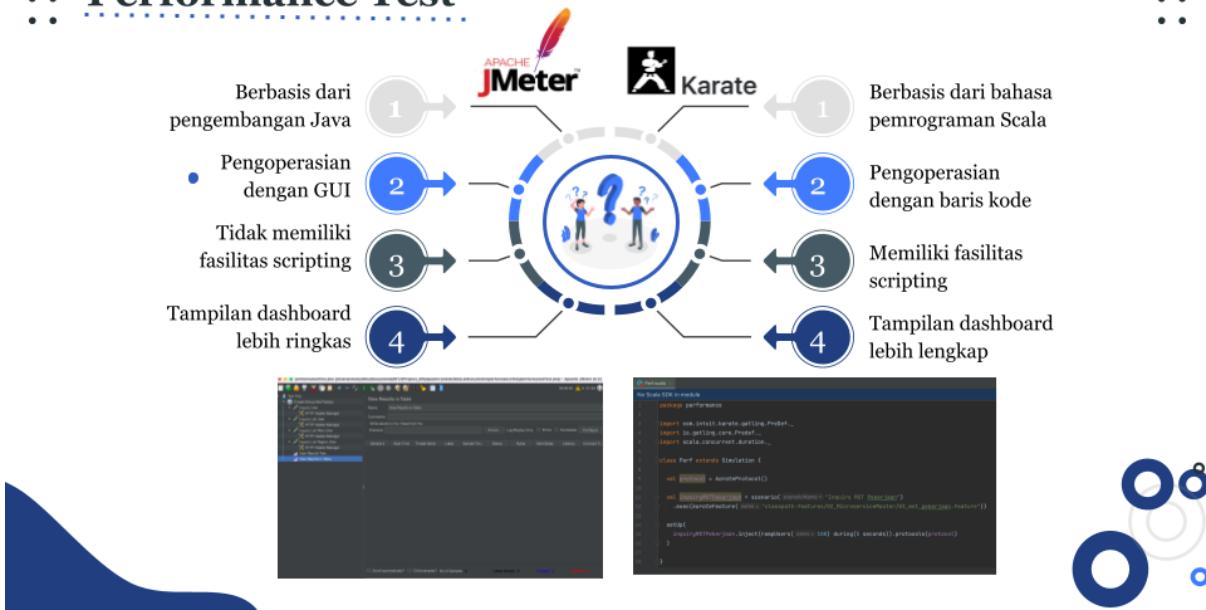
Penggunaan Sumber Daya



- Nilai Efisiensi Tools Karate
1. Penggunaan CPU : **20.35%**
 2. Jumlah Tread Aktif : **18.29%**
 3. Pemakaian Memori : **59.78%**



Performance Test



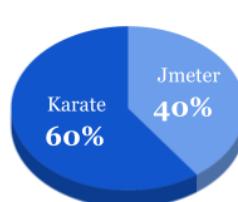
Performance Test



The slide compares the performance test results for JMeter and Karate, showing efficiency values and specific test details.

Tool	Test Details	Efficiency Value
JMeter	Menjalankan 100 request dalam waktu 15 second. Hasil dari performance test berbentuk dashboard, chart dan customs graphs. Run JMeter dapat dilakukan melalui GUI dari JMeter (CSV) dan Command Line untuk HTML.	40%
Karate	Menjalankan 100 request dalam waktu 6 second. Hasil dari performance test terdiri dari Global Information dan Detail Information. Run Karate Gatling via command line.	60%

Nilai Efisiensi Penggunaan Tools Karate

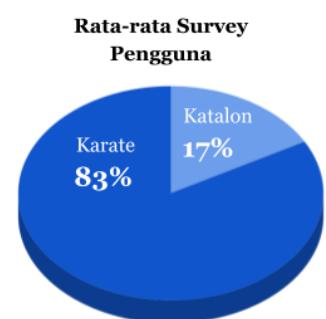
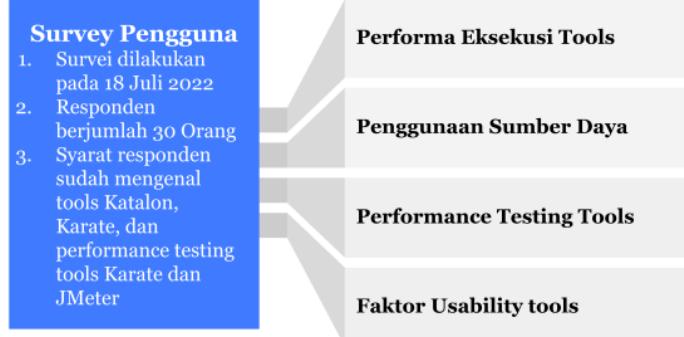


A pie chart titled "Nilai Efisiensi Penggunaan Tools Karate" comparing the efficiency of JMeter and Karate. The chart shows that Karate is 60% efficient and JMeter is 40% efficient.

Usability Tools



Usability Tools



Kesimpulan

Karate memiliki performa lebih baik pada sektor kecepatan eksekusi dan durasi pengujian

Karate menggunakan sumber daya yang lebih efisien pada sektor persentase penggunaan CPU, thread aktif, dan penggunaan memori(RAM)

Performance testing lebih mudah diimplementasikan menggunakan tools karate dibandingkan dengan tools JMeter

Karate lebih mudah diimplementasikan oleh pengguna non-programmer. Sedangkan Katalon studio lebih sesuai pada pengguna yang memiliki latar belakang programming



Terima Kasih

ABSTRAKSI

Pengelolaan pelayanan nasabah merupakan salah satu faktor penting yang diperhatikan oleh Bank BRI sebagai bank BUMN terbesar di Indonesia. Peningkatan pelayanan tersebut dapat dilakukan dengan memanfaatkan teknologi dan inovasi untuk mempermudah pelayanan nasabah. Penerapan teknologi terbaru dapat memberikan pengaruh terhadap produk-produk BRI yang digunakan untuk melayani nasabah secara langsung atau karyawan internal BRI.

Produk BRI yang digunakan untuk melakukan pelayanan perlu memiliki nilai kualitas dan kedayagunaan yang tinggi. Hal yang mempengaruhi kualitas produk teknologi tersebut salah satunya terletak pada proses *Quality Control*. Penerapan teknologi terbaru dapat dilakukan pada tahap pengujian perangkat lunak khususnya API dengan memanfaatkan Automation testing, sehingga pengujian perangkat lunak tersebut lebih konsisten dan efisien dibandingkan dengan pengujian manual.

Perangkat Automation testing yang saat ini digunakan di Bank BRI untuk melakukan pengujian API adalah Katalon Studio. Perangkat tersebut berjenis *licensed tools* sehingga memiliki fungsionalitas yang terbatas dan memiliki kelemahan dari sisi performa, sumber daya, dan struktur project yang kurang efisien. Berdasarkan hal tersebut, tools automation testing diperlukan pembaharuan khususnya dalam melakukan API Automation Testing. Salah satu tools yang dapat direkomendasikan yaitu tools Karate. Karate adalah salah satu framework untuk melakukan API Testing yang memiliki kode basis java dan menggunakan script BDD Cucumber pada proses implementasinya, BDD Cucumber merupakan script yang berbasis *natural language* sehingga dapat memudahkan pengguna non-programmer untuk dapat langsung memahami project struktur pada tools Karate.

Kata Kunci : API, Automation, Karate,Katalon

KATA PENGANTAR

Dengan mengucapkan puji syukur atas kehadiran Allah SWT yang telah memberikan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan makalah ini yang berjudul “**Analisis Perbandingan Integrasi Automation API Test Menggunakan Karate dan Katalon**”. Adapun makalah ini disusun untuk melengkapi syarat menyelesaikan pendidikan *Brilian Future Leader Program* (BFLP) IT PT Bank Rakyat Indonesia (Persero) Tbk. Pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini, di antaranya :

1. I Nyoman Sugiri Yasa selaku EVP Application Management and Operation Division (APP);
2. Aris Firmansyah selaku VP Quality & Improvement (QIM);
3. Fajar Ujian Sudrajat selaku VP SuperAppps (SAP)
4. Noviana Pramitasari selaku Team Leader E-Channel & From End Acceptance (EFA) dan pembimbing dalam mengerjakan makalah ini;
5. Imam Chanafi selaku Team Leader Loan Channel Platform Development (LCP);
6. Kedua Orang Tua penulis, Ibu Eti Rosyati dan Bapak Roswan Siliwangi (alm) serta Adikku tersayang Rosiana Nurfadjri Roswan yang telah memberikan dukungan secara moril maupun materil, selalu mendoakan penulis dengan tulus dan ikhlas hingga penulis dapat menyelesaikan Tugas Akhir ini.
7. Wirawan Noviana, M Iqbal Tawakal, Luthfi Kurniawan yang selalu memberikan dukungan, semangat, motivasi, doa, dan juga bantuannya kepada penulis agar Tugas Akhir ini dapat selesai tepat waktu. Terimakasih atas kesetiaan dan kesabarannya yang selalu diberikan kepada penulis
8. Seluruh anggota tim BRISPOT yang telah membantu dan memberikan informasi serta masukan dalam penyelesaian makalah ini;
9. Seluruh teman-teman BFLP IT Angkatan 8 tahun 2022 yang telah memberikan dorongan dan motivasi;
10. Seluruh pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu dalam penyusunan makalah ini.

Sebagai manusia biasa yang tak luput dari kesalahan, maka penulis meminta maaf atas segala kekurangan dan keterbatasan dalam penyusunan Tugas Akhir ini. Semoga apa yang terdapat dalam penulisan Tugas Akhir ini dapat bermanfaat bagi pembaca, dan perkembangan atau pemanfaatan teknologi informasi.

Jakarta, 25 Juni 2022

Reezky Illmawati

DAFTAR ISI

SURAT PERNYATAAN TIDAK PLAGIAT.....	ii
LEMBAR PERSETUJUAN PEMBIMBING MAKALAH.....	iii
FORMULIR BIMBINGAN MAKALAH.....	iv
PRESENTASI MAKALAH.....	vii
ABSTRAKSI.....	xv
KATA PENGANTAR.....	xvi
DAFTAR ISI.....	xviii
DAFTAR GAMBAR.....	xx
DAFTAR TABEL.....	xxi
DAFTAR GRAFIK.....	xxii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Rumusan Masalah.....	2
1.5 Tujuan Penelitian.....	2
1.6 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Software Testing.....	4
2.2 API Testing.....	6
2.3 Katalon Studio.....	7
2.4 Karate Framework.....	9
2.5 Struktur Project Tools Automation.....	10
2.5.1 Bahasa Script.....	10
2.5.2 Format file.....	11
2.5.3 Implementasi Library Java.....	11

2.5.4 Konfigurasi Project.....	12
2.5.5 Struktur Pengujian API Automation.....	12
2.6 BRISPOT.....	14
BAB III METODE PENELITIAN.....	15
3.1 Desain Penelitian.....	15
3.2 Perancangan Analisis.....	16
3.3 Teknik Analisis Perbandingan Data.....	16
3.3.1 Analisis Performa Eksekusi.....	16
3.3.2 Analisis Pemakaian Sumber Daya.....	17
3.3.3 Analisis Performance Test.....	17
3.3.4 Analisis Usability.....	17
BAB IV PEMBAHASAN.....	18
4.1 Objek Testing.....	18
4.2 Performa Eksekusi.....	18
4.3 Penggunaan Sumber Daya Tools Automation.....	26
4.4 Performance Test.....	28
4.5 Usability.....	32
4.5.1 Hasil Kuesioner.....	34
4.7 Hasil Pengujian.....	49
BAB V PENUTUP.....	51
5.1 Kesimpulan.....	51
5.2 Saran.....	51
DAFTAR PUSTAKA.....	xxiii

DAFTAR GAMBAR

Gambar 2.1. Format script BDD Cucumber pada Karate.....	10
Gambar 4.1 Tampilan Aplikasi JMeter.....	33
Gambar 4.2 Hasil Performance Test JMeter.....	34
Gambar 4.3 File Scala pada Performance Test Karate.....	35
Gambar 4.4 Hasil Performance Test Karate-Gatling.....	35
Gambar 4.5 Hasil dari Kuesioner Pertanyaan ke-1	40
Gambar 4.6 Hasil dari Kuesioner Pertanyaan ke-2.....	41
Gambar 4.7 Hasil dari Kuesioner Pertanyaan ke-3.....	42
Gambar 4.8 Hasil dari Kuesioner Pertanyaan ke-4.....	43
Gambar 4.9 Hasil dari Kuesioner Pertanyaan ke-5.....	44
Gambar 4.10 Hasil dari Kuesioner Pertanyaan ke-6.....	45
Gambar 4.11 Hasil dari Kuesioner Pertanyaan ke-7.....	46
Gambar 4.12 Hasil dari Kuesioner Pertanyaan ke-8.....	47
Gambar 4.13 Hasil dari Kuesioner Pertanyaan ke-9.....	48
Gambar 4.14 Hasil dari Kuesioner Pertanyaan ke-10.....	49

DAFTAR TABEL

Tabel 2.1 Bahasa Script Tools Karate dan Katalon.....	11
Tabel 2.2 Format File pada Tools Karate dan Katalon.....	12
Tabel 2.3 Implementasi Library Java pada Tools Automation API Testing.....	12
Tabel 2.4 Perbedaan Konfigurasi Project.....	13
Tabel 4.1 Hasil Test Microservice Auth.....	20
Tabel 4.2 Hasil Test Microservice Master.....	23
Tabel 4.3 Hasil Test Microservice Eksternal.....	25
Tabel 4.4 Hasil Test Microservice Role.....	27
Tabel 4.5 Hasil Test Microservice Office.....	29
Tabel 4.6 Perbandingan Penggunaan CPU.....	30
Tabel 4.7 Perbandingan Jumlah Thread Aktif.....	31
Tabel 4.8 Perbandingan penggunaan RAM.....	32
Tabel 4.9 Bobot Penilaian Jawaban Kuesioner.....	40
Tabel 4.10 Hasil Data Kuesioner	53
Tabel 4.11 Hasil Pengujian pada Katalon Studio dan Karate.....	55

DAFTAR GRAFIK

Grafik 4.1 Hasil Automation Microservice Auth.....	20
Grafik 4.2 Hasil automation Microservice Master.....	24
Grafik 4.3 Hasil automation Microservice Eksternal.....	26
Grafik 4.4 Hasil Automation Microservice Role.....	28
Grafik 4.5 Hasil automation Microservice Office.....	29

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT Bank Rakyat Indonesia (BRI) adalah salah satu bank milik pemerintah yang terbesar di Indonesia. Pada perkembangan teknologi di era digital, Bank BRI harus terus berinovasi untuk memenuhi kebutuhan nasabah sebagai prioritas utamanya. Bank BRI memiliki berbagai macam aplikasi untuk dikembangkan. Salah satunya aplikasi BRISPOT. Aplikasi BRISPOT adalah aplikasi untuk melakukan proses pengajuan fasilitas dan layanan kredit konsumen perbankan dari Bank BRI yang menawarkan solusi atas layanan kredit perbankan berbasis digital.

Layanan bank BRI yang berbasis digital dipengaruhi oleh kualitas aplikasi. Salah satu cara untuk menentukan kualitas aplikasi dengan melakukan *automation testing*. *Automation testing* adalah sebuah metode pengujian *software* yang dilakukan oleh seorang quality assurance(QA) untuk menguji dan membandingkan antara output yang dihasilkan oleh aplikasi dengan output yang diharapkan dengan menggunakan perangkat lunak khusus. Proses *Automation testing* dirancang untuk memberikan efisiensi, efektivitas, dan akurasi yang lebih tinggi.

Objek automation testing pada dasarnya terbagi menjadi tiga bagian yaitu Web Testing, Mobile Testing, dan API Testing. Pada penelitian ini dijelaskan mengenai API *automation testing* yang saat ini digunakan di Bank BRI menggunakan tools katalon studio. API Testing memegang peranan yang penting dalam hal pengelolaan data, dilihat dari segi kompleksitas dan performance yang lebih stabil dibandingkan dengan automation berbasis UI test. Penggunaan katalon studio untuk melakukan API *testing* pada Aplikasi BRISPOT memiliki kelemahan pada sisi performance test, penggunaan sumber daya, dan struktur project yang kurang efisien. Oleh karena itu, perangkat automation testing yang saat ini digunakan perlu diberikan pembaharuan. Salah satu pilihan yang dapat diimplementasikan pada proses API Automation Testing adalah Karate.

API Automation Testing menggunakan karate memiliki kelebihan dibandingkan dengan katalon studio dalam hal performance, sumber daya, dan struktur kode yang lebih efisien. Tools karate bersifat open source dan memiliki struktur kode yang berbasis java.

Penulisan script pada karate menggunakan BDD Cucumber dengan bahasa gherkin seperti natural language.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang di atas, dapat dilakukan identifikasi masalah dalam penelitian ini sebagai berikut:

1. Penggunaan tools Katalon Studio membutuhkan resource yang besar
2. Struktur project pada tools Katalon Studio sangat kompleks
3. Proses eksekusi testing membutuhkan waktu yang lama pada tools Katalon Studio
4. Katalon studio yang open source tidak memiliki fitur performance test

1.3 Batasan Masalah

Batasan masalah bertujuan agar penelitian yang dilakukan lebih terarah dan relevan.

Adapun batasan masalah dalam penelitian sebagai berikut:

1. Proses API Automation testing menggunakan tools Katalon dan Karate
2. Project yang dilakukan proses testing adalah Microservice Brispot Phase III
3. Hasil analisis menggunakan reporting dari kedua tools dalam format html
4. Performance testing dijalankan pada tools Karate menggunakan library Gatling dan tools eksisting BRI menggunakan Jmeter
5. Project pada kedua tools dijalankan secara local

1.4 Rumusan Masalah

Rumusan masalah yang dibahas pada penelitian ini sebagai berikut:

1. Bagaimana hasil dari performa eksekusi menggunakan Katalon Studio dan Karate?
2. Bagaimana penggunaan sumber daya menggunakan Katalon Studio dan Karate?
3. Bagaimana struktur project pada tools Katalon Studio dan Karate?
4. Bagaimana hasil performance test pada tools Karate dan Jmeter?
5. Bagaimana hasil dari pengujian pada tools Katalon Studio dan Karate?

1.5 Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut:

1. Menganalisis perbandingan hasil dari performa eksekusi pada tools Katalon Studio dan Karate
2. Mengetahui hasil sumber daya yang diperlukan pada tools Katalon Studio dan Karate
3. Mengetahui perbandingan tingkat usability pada tools Katalon Studio dan Karate
4. Mengetahui hasil dari performance test API Automation Test pada tools Karate dan Jmeter

5. Memberikan rekomendasi tools automation untuk pengujian aplikasi di BRI

1.6 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Meningkatkan efektivitas dalam melakukan pengujian pada Automation API
2. Hasil dalam penelitian ini, dapat dijadikan pertimbangan sebagai pilihan penggunaan tools lain untuk melakukan pembaharuan pada proses Automation API di Bank BRI
3. Hasil reporting yang *di-generate* oleh Karate lebih jelas, lebih singkat, dan lebih interaktif sehingga lebih mudah dipahami.

BAB II

TINJAUAN PUSTAKA

2.1 Software Testing

Software Testing adalah suatu proses, untuk mengevaluasi fungsionalitas aplikasi perangkat lunak dalam mendeteksi apakah perangkat lunak yang dikembangkan telah berfungsi dengan baik dan memenuhi persyaratan yang ditentukan, serta mengidentifikasi cacat pada aplikasi. Tujuan dasar pengujian perangkat lunak untuk menghilangkan bug dan meningkatkan berbagai aspek perangkat lunak, seperti kinerja, pengalaman pengguna, keamanan, dan sebagainya dalam menghasilkan produk yang berkualitas. Metode pengujian aplikasi dibagi menjadi 2 kategori sebagai berikut :

1. Manual Testing

Metode pengujian aplikasi yang dilakukan secara manual, dioperasikan dan diamati oleh indera manusia secara langsung. Proses ini dilakukan pada saat pengujian automation tools tidak dapat dijalankan karena membutuhkan human verification. Pengujian secara manual dilakukan dengan cara memvalidasi fitur utama, menjalankan test case dan membuat laporan ringkas perihal skenario yang telah dijalankan tanpa menggunakan tools automasi. Metode ini memiliki kelebihan dari hasil pengujian yang akurat berdasarkan feedback dari pengguna. Sedangkan kelemahan dari pengujian secara manual adalah rentan terdapat kesalahan pada prosedur testing karena ketidaktelitian dan memungkinkan terjadi inkonsistensi sehingga hasil pengujian menjadi tidak valid.

2. Automation Testing

Metode pengujian secara otomatis adalah proses pengujian yang dilakukan dengan menjalankan program tertentu secara otomatis untuk mensimulasikan penggunaan aplikasi oleh pengguna. Automation testing adalah sebuah teknik dalam proses pengujian aplikasi dengan memanfaatkan alat berupa perangkat lunak khusus untuk menguji dan membandingkan output yang dihasilkan dengan output yang diharapkan. Automation testing dapat diterapkan pada produk-produk perangkat lunak yang sedang dikembangkan seperti, Mobile Application, Web Application, REST API, Performance Test. Perancangan automation testing dilakukan untuk memberikan efisiensi, efektivitas, dan akurasi yang tinggi pada proses pengujian perangkat lunak. Cara kerja automation testing dengan menjalankan script yang sudah ditentukan alur dan output yang diharapkan dengan melakukan validasi terhadap respon atau data yang ditampilkan berdasarkan script tersebut. Kelebihan

dari pengujian automation memiliki konsistensi yang tinggi dan dapat dilakukan secara terus-menerus pada prosedur testing yang memiliki kompleksitas tinggi. Kelemahan dari metode automation testing ini adalah prosedur testing bergantung pada script yang dibuat oleh Test Engineer untuk dijalankan melalui platform testing.

Pengujian perangkat lunak merupakan suatu hal yang sangat penting dalam memastikan kualitas, kepuasan pelanggan, mengurangi biaya pengembangan, dan meningkatkan keamanan pada aplikasi. Berikut beberapa karakteristik dari aplikasi yang berkualitas seperti Correctness, Reliability, Efficiency, Integrity, Usability, Maintainability, Flexibility, Testability, Portability, Reusability. Jenis-jenis dari software testing sebagai berikut :

1. Unit testing berfungsi untuk memvalidasi bahwa setiap unit perangkat lunak berfungsi seperti yang diharapkan. Unit adalah komponen terkecil dari aplikasi yang dapat diuji, dimana pengujian dilakukan pada bagian basic dari kode program, contohnya pada pengujian kode program pada event, procedure dan function. Dengan Unit testing meyakinkan bahwa masing-masing unit bekerja sebagaimana mestinya. Unit Testing biasanya dilakukan oleh seorang software developer
2. Integration testing berfungsi untuk memastikan bahwa komponen atau fungsi perangkat lunak yang dikembangkan beroperasi sesuai dengan tujuan secara berbarengan dengan komponen lain.
3. System testing Mencakup testing aplikasi yang telah selesai dikembangkan. Karena itu, aplikasi harus terlihat dan berfungsi sebagaimana mestinya terhadap end-user atau pengguna akhir. Untuk itu, pengujian dilakukan menggunakan data yang menggambarkan pengguna sesungguhnya terhadap aplikasi.
4. Acceptance testing berfungsi untuk memverifikasi apakah seluruh sistem berfungsi sebagaimana dimaksud, proses ini biasanya dilakukan oleh business owner atau user. Dalam beberapa proyek, acceptance testing dilakukan pada putaran terakhir proses pengujian yaitu sebelum sistem diserahkan kepada user. Acceptance Testing biasanya dilakukan setelah rangkaian testing seperti Unit Testing, Integration Testing, dan System Testing selesai dan menggunakan metode Black Box Testing, dengan menggunakan dokumen test case untuk di presentasikan di akhir ke user / client.
5. Regression testing berfungsi untuk memeriksa apakah fitur baru merusak atau menurunkan fungsionalitas. Pengujian kewajaran dapat digunakan untuk memverifikasi menu, fungsi, dan perintah di tingkat permukaan, ketika tidak ada

waktu untuk melakukan pengujian secara menyeluruh. jenis pengujian ini dilakukan saat mengembangkan software untuk mengetahui apakah fungsional sistem berjalan dengan baik. Singkatnya, tujuan utamanya yaitu untuk meminimalisir bug yang mungkin akan muncul setelah adanya pembaharuan fitur pada software.

6. Performance testing berfungsi untuk menguji kinerja perangkat lunak di bawah beban kerja yang berbeda. Pengujian beban, misalnya, digunakan untuk mengevaluasi kinerja di bawah kondisi beban kehidupan nyata. Jenis testing ini merupakan perpaduan antara integration dan usability test yang menentukan apakah sistem dapat memenuhi kriteria kinerja berbasis waktu seperti response time atau throughput.
7. Stress Testing, berfungsi untuk menguji seberapa besar tekanan yang dapat diambil sistem sebelum gagal. Dianggap sebagai jenis pengujian non-fungsional. contoh pengujian yang biasanya dilakukan dalam membuat sebuah website, dimana stress testing dilakukan untuk mengetahui sekuat apa server website dapat menampung visitor dalam website tersebut, dengan cara melakukan akses dengan jumlah banyak diwaktu yang bersamaan menggunakan bantuan tools
8. Usability Testing, berfungsi untuk memvalidasi seberapa baik pengguna dapat menggunakan aplikasi atau web untuk menyelesaikan kebutuhannya. Selain itu, testing ini juga berfungsi untuk mengevaluasi suatu produk dengan mengujinya pada pengguna.

2.2 API Testing

API Testing merupakan pengujian kumpulan API untuk memeriksa fungsionalitas dalam mengembalikan respons yang sesuai, memeriksa respons berdasarkan parameter input (request), dan memeriksa berapa lama waktu yang dibutuhkan API untuk mengambil dan mengotorisasi data juga.. API testing dapat memberikan gambaran tentang bagaimana response dari aplikasi web, sebelum GUI dari aplikasi dikembangkan.

API (*Application Programming Interface*) adalah seperangkat aturan yang berisi metode komunikasi yang jelas untuk berinteraksi satu sama lain. API memiliki cara kerja yang dilakukan aplikasi mengakses API, API melakukan request ke server, server memberikan respon ke API, dan API menyampaikan respon ke aplikasi. Beberapa jenis dari API sebagai berikut :

1. Public API (Open API)

Public API adalah salah satu jenis API yang dapat digunakan secara umum dalam lintas platform. Contohnya, API untuk Google Maps dan API untuk data Covid Indonesia.

2. Private API

Private API adalah API yang tidak terbuka untuk penggunaan umum. Private API biasanya digunakan untuk keperluan internal dalam pengembangan aplikasi tertentu. Contohnya, API dari back end yang digunakan untuk mengakses front end dari sebuah website atau pengembangan aplikasi mobile.

3. Partner API

Partner API digunakan untuk kepentingan umum yang memiliki izin penggunaannya sesuai dengan perjanjian yang berlaku untuk diterapkan pada aplikasi tertentu.

4. Composite API

Composite API merupakan API yang menyimpan data dari berbagai server atau hosting dalam satu tempat. Manfaat dari penggunaan composite API, pengguna dapat memperoleh berbagai jenis data yang diperlukan dalam sekali akses.

Arsitektur pada API terdiri menjadi 3 bagian sebagai berikut:

1. RPC merupakan arsitektur dari API untuk membuat komunikasi antara client side dan server side yang dilakukan dengan konsep sederhana. RPC memiliki dua jenis, yaitu XML-RPC dan JSON-RPC. XML-RPC menggunakan format XML sebagai media perpindahan data, sedangkan JSON-RPC menggunakan JSON untuk perpindahan data.
2. SOAP (*Simple Object Access Protocol*) adalah kategori API yang memiliki format XML
3. REST adalah kategori API yang memiliki format JSON. Proses permintaan dan penerimaan data menggunakan protokol HTTP dengan verbs GET, POST, PUT dan DELETE dalam berinteraksi.

2.3 Katalon Studio

Katalon studio merupakan platform automasi berbasis selenium yang berfungsi untuk menjalankan proses secara automasi. Tool tersebut merupakan aplikasi open source untuk melakukan pengujian secara otomatis yang telah dikembangkan oleh Katalon LLC yang dapat dijalankan di semua sistem operasi seperti Windows, MAC os dan linux. Katalon Studio menggunakan antarmuka IDE khusus untuk melakukan pengujian. Katalon Studio memiliki tiga fitur utama untuk melakukan beberapa pengujian di beberapa platform seperti Web testing, API testing, dan Mobile testing. Selain itu Katalon studio telah terintegrasi dengan beberapa teknologi dari luar seperti github. Katalon Studio sendiri telah menyediakan UI grafis yang beragam tampilan, menu, pohon tabel dan lain-lain untuk mengelola test case objek dan file data. Katalon studio

mengintegrasikan semua komponen yang diperlukan dengan keyword atau kata kunci bawaan dan template proyek ke dalam automation framework atau kerangka kerja otomasi lengkap. Katalon studio mudah digunakan oleh pemula tetapi memberikan kemampuan tingkat lanjut untuk pengguna ahli atau expert user.

Secara garis besar, Katalon dirancang untuk membuat dan menggunakan kembali skrip pengujian otomatis untuk UI tanpa pengkodean. Katalon Studio memungkinkan menjalankan pengujian otomatis elemen yang terdapat pada platform yang dipilih, dengan menyediakan berbagai macam komponen, build-in keyword, dan template yang dapat digunakan untuk keperluan pembuatan test case. Katalon itu sendiri dapat digunakan di Microsoft Windows, macOS, dan Linux. Katalon memiliki beberapa test artifact yang berfungsi untuk mempermudah proses alur kerja pengguna, diantaranya adalah sebagai berikut

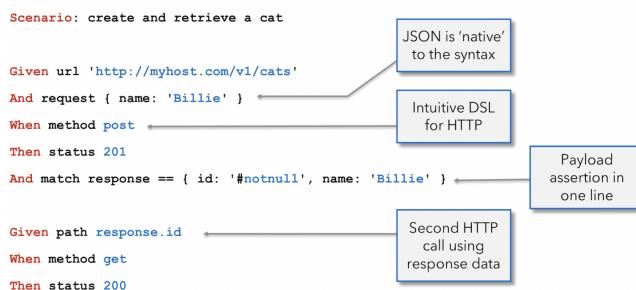
1. Profile adalah list daftar global variabel yang dikembangkan pada proses testing.
2. Test Case adalah langkah-langkah skenario pengetesan yang disusun menjadi satu kesatuan test skenario. Proses membuat test case dapat dilakukan dengan cara melakukan proses record, manual, atau script testing.
3. Object Repository adalah kumpulan objek atau locator dari program yang disimpan pada saat dilakukan proses recording ataupun spy object pada mobile atau web. Object repository juga tempat dimana untuk mendefinisikan web service request pada katalon dengan format HTTP request.
4. Test Suites adalah kumpulan dari test case atau test case collection dari project yang sedang dikerjakan. Proses untuk mengenerate report dilakukan dengan menjalankan test suites. Pada test suites juga memiliki variabel data binding untuk menghubungkan variabel pada test case dengan data file yang didefinisikan pada test data.
5. Data Files adalah test data yang digunakan untuk project tersebut dengan format CSV atau Excel file
6. Checkpoint adalah daftar checkpoint atau snapshot dari data uji yang diambil pada waktu tertentu.
7. Keyword adalah kata kunci custom yang telah ditetapkan pada saat pembuatan test case, kata kunci tersebut didefinisikan oleh user itu sendiri sesuai dengan kebutuhan Selain Katalon Studio, Katalon memiliki beberapa produk lainnya seperti Katalon Recorder, Katalon TestOps, Katalium, Katalon Studio Enterprise, Katalon Runtime

Engine, Katalon TestOps Enterprise dan yang akan segera launching ada Katalon CloudRun, Advanced Reports, AI Optimizations dan Performance Testing.

2.4 Karate Framework

Karate adalah tools yang bersifat Open Source untuk melakukan automation testing khususnya pada REST API Testing. Penulisan struktur kode pada Karate menggunakan bahasa Java. Sedangkan dalam proses Implementasi pengujian otomatisasi menggunakan Behaviour Driven Development (BDD) Gherkin syntax yang lebih mudah dipahami (Natural Language). Proses pengujian pada karate dapat dilakukan secara API Testing, Mocking, Web UI Testing, Performance Testing (Gatling).

Karate memiliki struktur project yang mudah dipahami seperti file karate config merupakan file yang mengintegrasikan seluruh function pada API Testing, File pom.xml merupakan file untuk menambahkan plugin dan dependencies untuk keperluan API Testing. Sedangkan jika menambahkan script automation API dengan membuat file feature yang ditulis dengan format script BDD sebagai berikut :



Gambar 2.1. Format script BDD Cucumber pada Karate

Pada gambar 2.1 menampilkan detail dari format script BDD Cucumber pada karate yang dijalankan untuk proses API Automation Testing. Detail dari Format script BDD Cucumber sebagai berikut :

1. Feature adalah file yang berisi fitur API Testing yang dikembangkan oleh pengguna.
2. Background adalah file yang mendefinisikan payload atau file lainnya yang diperlukan untuk keperluan testing.
3. Scenario adalah file yang terdiri dari script BDD Cucumber untuk menjalankan proses testing sebagai berikut :
 - Given : pre-condition yang mendefinisikan bash_url
 - When : Action yang dapat dijalankan setelah mendefinisikan proses Given. Action ini mendefinisikan method yang digunakan untuk request API seperti GET, POST, PUT, DELETE, dan lain sebagainya.

- And : Action yang dapat mengkombinasikan dua atau lebih action. Biasanya digunakan untuk mendefinisikan path, menampilkan hasil respon dari proses API Testing atau memverifikasi respon
- Then : Proses untuk mendefinisikan status code request dari API Testing

2.5 Struktur Project Tools Automation

Struktur project tools automation menjelaskan format dan tipe file yang digunakan untuk menyusun sebuah script automation testing dan menjelaskan keterkaitan antar file tersebut. Penjelasan tersebut dilakukan komparasi antara tools Katalon Studio dan Karate. Katalon Studio dijalankan pada IDE Katalon Studio sedangkan Karate dijalankan pada IDE IntelliJ IDEA. Perbandingan struktur project terbagi menjadi empat jenis yaitu berdasarkan bahasa script, format file pada test case, implementasi library Java, konfigurasi project.

2.5.1 Bahasa Script

Terdapat perbedaan bahasa script yang digunakan, pada katalon studio menggunakan framework Groovy. Katalon menggunakan Groovy sebagai engine untuk script mereka. Seperti yang kita tahu, Groovy merupakan bahasa yang memperluas bahasa pemrograman Java dengan bentuk yang lebih simple dan lebih mudah untuk dimengerti.

Tabel 2.1 Bahasa Script Tools Karate dan Katalon

Karate	Katalon
 Gherkin	 Groovy

Tabel 2.1 menjelaskan bahwa format Groovy mengimplementasi script dari java dan turunannya serta memiliki fungsi-fungsi dasar yang diadopsi dari java. pada karate framework yang digunakan adalah Gherkin, framework tersebut memiliki bentuk sintaks natural language. Pada framework gherkin memiliki empat instruksi utama yaitu “Given”, “When”, “Then”, dan “And”.

2.5.2 Format file

Perbedaan berikutnya terdapat pada format file yang digunakan untuk menulis script testing. Format tersebut direpresentasikan dengan ekstensi file yang digunakan. Terkait dengan pembahasan 2.5.1 mengenai script yang digunakan, karate menggunakan script Gherkin yang memiliki ekstensi file test case yaitu *.feature, sedangkan katalon studio menggunakan script dari groovy yang memiliki ekstensi file test case yaitu *.tc. Ekstensi tersebut hanya dapat dioperasikan oleh aplikasi katalon studio, sehingga katalon studio memiliki karakteristik “*closed environment*”.

Tabel 2.2 Format File pada Tools Karate dan Katalon

Karate	Katalon
*.feature	*.tc

Tabel 2.2 menjelaskan tentang perbedaan ekstensi file yang digunakan dalam pembuatan test case. Karate menggunakan ekstensi file yaitu *.feature, penggunaan ekstensi tersebut membuat karate menjadi lebih fleksibel dan dapat dibuka dengan text editor umum seperti Visual Studio Code dan IntelliJ IDEA. Pada katalon ekstensi yang digunakan adalah *.tc, ekstensi tersebut hanya dapat dibuka secara optimal dengan menggunakan aplikasi katalon studio, faktor tersebut menjadikan katalon memiliki nilai fleksibilitas yang lebih rendah jika dibandingkan dengan karate.

2.5.3 Implementasi Library Java

Kedua tools automation API testing ini mengimplementasi library java pada operasionalnya. Pada karate library java digunakan pada saat penulisan script runner. Runner adalah sebuah file yang menjalankan lebih dari satu test case, sedangkan pada katalon studio library java digunakan sebagai dasar dari penggunaan script groovy. penggunaan tersebut meliputi konsep OOP (*Object Oriented Programming*), fungsi dasar seperti operasi string, fungsi aritmatik dan lainnya.

Tabel 2.3 Implementasi Library Java pada Tools Automation API Testing

Karate	Katalon
Menggunakan Java sebagai class untuk membuat runner feature	Menggunakan basis java untuk implementasi framework Groovy

Tabel 2.3 menjelaskan tentang perbedaan implementasi library java pada kedua tools automation API testing. Perbedaan tersebut memiliki pengaruh pada script yang ditulis.

Pada katalon studio library java digunakan hanya sebagai konsep acuan dalam penulisan script groovy, sehingga bentuk penulisan memiliki perbedaan dengan bahasa java native. Pada karate library java digunakan secara utuh untuk menjalankan skenario yang telah ditulis, hal tersebut dibuktikan dengan file runner yang memiliki ekstensi *.java sehingga format penulisan memiliki kesamaan dengan java native.

2.5.4 Konfigurasi Project

Konfigurasi project pada tools automation API testing berfungsi untuk mendefinisikan kebutuhan, *environment*, dan *life-cycle* pada aplikasi. Kebutuhan yang dimaksud salah satunya adalah penggunaan library *eksternal*.

Pada karate konfigurasi project terdapat pada sebuah file “karate-conf.js” yang diletakan pada direktori *root*. File tersebut tidak terdapat secara default pada saat pembuatan project karate, user perlu menambahkan sendiri file konfigurasi tersebut. File karate-conf.js berisi nilai-nilai variabel yang digunakan secara global pada saat menjalankan testing seperti waktu *timeout*, *base url*, penggunaan jenis environment yang digunakan, dan lain-lain.

Pada katalon studio konfigurasi dilakukan pada menu project structure yang terdapat sebagai bawaan dari aplikasi katalon studio, sehingga pengaturan yang digunakan pada saat menjalankan testing lebih tertata dan berbasis GUI.

Tabel 2.4 Perbedaan Konfigurasi Project

Karate	Katalon
Menggunakan file berbasis javascript(karate-conf.js) untuk mengatur konfigurasi project	Memiliki options menu terpisah berbasis UI untuk mengatur konfigurasi project

Pada tabel 2.4 diatas dijelaskan tentang perbedaan struktur konfigurasi pada project. Pada katalon studio konfigurasi dilakukan berbasis GUI sehingga memiliki nilai user friendly yang lebih baik dibandingkan dengan karate, pada karate konfigurasi project perlu ditambahkan secara manual oleh user, sehingga memiliki potensi kesalahan penulisan yang dibuat oleh user tersebut.

2.5.5 Struktur Pengujian API Automation

Proses pengujian API Testing pada setiap tools memiliki langkah yang berbeda-beda.

Pada katalon studio, API Testing dilakukan dengan menjalankan script sebagai berikut :

Tools Katalon Studio :

```

GlobalVariable.pn = '00182952'
GlobalVariable.password = 'P@ssw0rd'
REQUEST = '''{
    "username": "${pn}",
    "password": "${password}",
    "app": "Brispot_Mobile_Pemrakarsa",
    "version_app": "300070006",
    "latitude": "",
    "longitude": ""
}'''
CustomKeywords.'microservice.api.sendRequestNew'(REQUEST)
CustomKeywords.'microservice.response.verifyResponseCode'('00')

```

1. Import Keyword yang digunakan untuk keperluan testing
2. Menginisiasi variabel
3. Generate request body
4. Melakukan HTTP Request pada API yang ditentukan dengan membawa request body
5. Menampilkan hasil dari HTTP Request
6. Memformat ulang hasil dari HTTP Request tersebut dalam bentuk JSON
7. Memverifikasi response code dari hasil HTTP Request

Sedangkan pada tool karate mengikuti gaya "Given When Then And" untuk skrip pengujian yang membantu dalam memahami dan memetakan persyaratan sebagai berikut :

```

Scenario: Login sebagai Mantri
Given url baseUrl
And path 'v1/login'
And headers headerVerif
And request payloadUserMantri
And method POST
Then status 200
And print 'response---', response
* def tokenMicro = 'Bearer ' + response.responseData.token
And print tokenMicro

```

```
And match response == expectedOutputLogin
```

1. Komponen pertama adalah kata kunci yang mendefinisikan URL/ End point API.
2. Mendefinisikan 'param' dalam bentuk json
3. Mendefinisikan, 'header', yang merupakan cara untuk pengujian API.
4. Metode yang diperlukan untuk HTTP request API.
5. Validasi kode status dan waktu respons
6. Mencetak respons
7. Verifikasi respon yang diberikan

2.6 BRISPOT

BRISPOT merupakan salah satu produk dari Bank Rakyat Indonesia yang digunakan untuk memenuhi kebutuhan nasabah dalam hal menawarkan berbagai macam produk pinjaman atau kredit. BRISPOT memiliki beberapa jenis aplikasi yang tersebar melalui dua buah *platform* yaitu web dan mobile. Berikut beberapa jenis aplikasi pada aplikasi BRISPOT sebagai berikut:

1. BRISPOT Pemrakarsa, aplikasi ini diperuntukan kepada RM Kecil dan Mantri, berfungsi untuk melakukan prakarsa pengajuan kredit KUR, Mikro, dan Ritel
2. BRISPOT Pemutus, aplikasi ini diperuntukan untuk melakukan putusan Kredit yang telah selesai proses prakarsa dan berstatus menunggu putusan, dan hanya bisa digunakan oleh pekerja dengan Level Manajerial sebagai Pemutus, seperti AMP, MP, dan Pinca
3. BRISPOT Konsumen, aplikasi ini diperuntukan kepada RM untuk melakukan pengajuan produk pinjaman seperti KPR, Briguna, dan Kartu Kredit, mulai dari prakarsa kredit, hingga putusan kredit
4. BRISPOT Konsumen Web, Web ini masih berhubungan dengan aplikasi BRISPOT Konsumen, berfungsi untuk berbagai keperluan khususnya monitoring pengajuan kredit, menambahkan data referral kredit, melakukan putusan (khusus Briguna)..
5. BRISPOT *Online Loan Application* atau yang lebih dikenal dengan BRISPOT Eksternal, aplikasi ini diperuntukan untuk Nasabah atau Calon Nasabah, berfungsi untuk melakukan pengajuan kredit seperti Briguna, KPR, dan Mikro/Ritel secara mandiri.

BAB III

METODE PENELITIAN

Metode penelitian pada penulisan ini menggunakan metode eksperimental. Metode tersebut memiliki beberapa variabel untuk melakukan pengujian secara sistematis seperti Desain penelitian, Perancangan Analisis, Teknik Analisis Perbandingan Data.

3.1 Desain Penelitian

Desain penelitian dilakukan dengan merancang alur pengujian dalam 2 platform automation testing yaitu katalon studio dan karate. Alur pengujian tersebut antara lain :

1. Menentukan objek penelitian.
2. Membuat scenario test dan data testing pada tools API Automation yaitu Katalon dan Karate dengan menggunakan project BRISPOT Microservice Phase III sebagai target pengujian.
3. Membuat script API Automation pada tools Katalon dan Karate
4. Menjalankan hasil proses testing pada tools Katalon dan Karate
5. Menganalisis perbandingan yang terdapat pada tools Katalon dan Karate berdasarkan proses dan hasil pengujian.
6. Menganalisis hasil *performance testing* pada tools yang tersedia di Karate dan tools eksisting yang biasa dilakukan oleh BRI menggunakan software JMeter.

3.2 Perancangan Analisis

Perancangan Analisis pada penelitian ini dilakukan dengan membandingkan dua tools automation API testing yaitu Katalon Studio dan Karate. Hasil dari perbandingan tersebut diukur dan dikelompokkan ke dalam lima parameter, yaitu :

1. Performa

Performa dari kedua platform tersebut adalah satuan waktu yang dibutuhkan untuk mengeksekusi alur pengujian pada http request. Satuan waktu yang digunakan adalah *milisecond*. Pengukuran performa memiliki sifat efisiensi yaitu semakin singkat waktu yang dibutuhkan untuk menyelesaikan alur pengujian, maka semakin baik.

2. Penggunaan Sumber Daya

Sumber daya yang diukur adalah persentase penggunaan CPU, thread aktif pada saat menjalankan automation testing, dan alokasi memori(RAM). Tiga komponen tersebut dijadikan parameter pengukuran karena pengelolaan sumber daya pada perangkat

sangat bergantung pada seberapa efektif tools automation testing tersebut memaksimalkan performa dengan penggunaan sumber daya seminimal mungkin.

3. Struktur Project

Struktur project diperlukan untuk mengetahui perbandingan kompleksitas pada tools katalon studio dan karate. Tolak ukur yang digunakan untuk membandingkan kedua tools automation tersebut adalah format file yang digunakan dan contoh penggunaan file tersebut untuk menjalankan sebuah pengujian API.

4. Performance Test

Performance test merupakan bentuk pengujian pada API dengan melakukan http request pada API yang dituju. request tersebut dilakukan dengan jumlah yang tinggi untuk mensimulasikan kondisi sebenarnya pada saat jumlah pengguna yang masif melakukan request secara bersamaan. Performance test bertujuan untuk mengetahui performa sebuah API yang diukur dalam satuan waktu. Performance test memiliki sifat efektivitas yaitu semakin rendah waktu yang dibutuhkan dalam proses request maka semakin baik API tersebut.

5. Usability

Usability adalah hasil analisis mengenai seberapa efisien dan efektif tools automation testing saat digunakan untuk melakukan tugasnya yaitu menjalankan proses automation testing. Hal tersebut dilakukan untuk mengetahui kemungkinan kendala penggunaan tools API Automation testing yang diimplementasikan pada project.

3.3 Teknik Analisis Perbandingan Data

Teknik yang dilakukan untuk membandingkan dua platform automation API testing adalah dengan melakukan pengamatan pada parameter yang telah disebutkan pada perancangan analisis, diantaranya adalah:

3.3.1 Analisis Performa Eksekusi

Analisis performa dilakukan dengan menjalankan alur pengujian API pada kedua platform. Hasil dari masing-masing platform akan dibandingkan waktu eksekusinya berdasarkan langkah-langkah yang dilakukan oleh tools tersebut. Data yang diambil merupakan data dengan satuan waktu *milisecond(ms)* dan diambil dari report yang di-*generate* secara otomatis setelah tools tersebut selesai menjalankan alur pengujian yang sama.

3.3.2 Analisis Pemakaian Sumber Daya

Analisis sumber daya dilakukan dengan mengamati persentase penggunaan CPU, thread processor yang sedang aktif, dan alokasi memori(RAM) pada saat dua tools automation tersebut dijalankan. Data tersebut didapatkan dari tampilan status pada activity monitor di perangkat yang sedang digunakan.

3.3.3 Analisis Performance Test

Analisis performance test dilakukan dengan mengkomparasi tools existing yang saat ini digunakan oleh BRI untuk menguji performa API yaitu JMeter dengan library terkait Karate untuk menguji performa API yaitu Gatling. Keduanya akan menjalankan perintah dan skenario yang sama untuk melakukan performance test. Hasil dari komparasi tersebut akan dibandingkan berdasarkan waktu eksekusi dalam satuan *milisecond*.

3.3.4 Analisis Usability

Analisis Usability dilakukan dengan mendefinisikan kelebihan, kekurangan, dan faktor implementasi pada project BRI yang sedang berjalan saat ini dan yang akan datang. Hasil dari perbandingan tersebut menentukan faktor kebergunaan tools automation API testing mana yang lebih baik untuk digunakan pada pengujian API di BRI.

BAB IV

PEMBAHASAN

4.1 Objek Testing

Objek yang digunakan dalam automation API test ini adalah API dari project Microservice Phase III. Microservice Phase III adalah Project yang dikembangkan pada Brispot Mobile & Web untuk meningkatkan kemudahan, reliabilitas, dan keamanan dalam penggunaan aplikasi BRISPOT. Selain itu, pada project ini terdapat pengembangan ESB Core untuk memberikan reliabilitas yang optimal dan fitur audit trail lebih baik. Sistem PGS BRISPOT yang terintegrasi dengan platform milik SDM BRI dan Mapping Role dalam memberikan kemudahan pengelolaan menu untuk setiap segmen/proses bisnis pada BRISPOT.

4.2 Performa Eksekusi

Perbandingan performa dari kedua tools automation testing dilakukan dengan menjalankan beberapa alur automation yang di dalamnya terdapat beberapa metode yaitu http request, pengecekan response, pencocokan response yang diharapkan dan *generate report*. Alur pengujian yang digunakan diantaranya adalah:

1. Microservice Auth

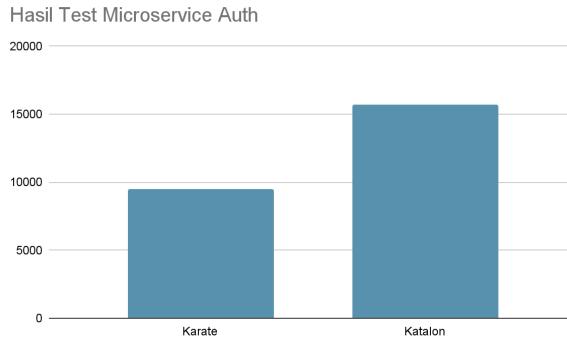
Pada Microservice Auth, skenario yang dijalankan adalah :

- | | |
|--|------------------|
| - Login sebagai Mantri | - Logout Brispot |
| - Login sebagai RM | - Check Session |
| - Login sebagai Kepala Unit | - Check Access |
| - Login sebagai Pimpinan Cabang | - Create Session |
| - Validasi kondisi “User anda sedang login” | - Request OTP |
| - Validasi kondisi “User atau Password tidak sesuai” | - Validate OTP |
| - Kondisi “No Parameter” | - Create Tester |
| | - Check Tester |
| | - Check Version |
| | - Set Version |

Dari skenario tersebut didapatkan hasil yang ditampilkan pada tabel 4.1 berikut

Tabel 4.1 Hasil Test Microservice Auth

Berdasarkan hasil test yang ditampilkan pada tabel 4.1 di atas diketahui bahwa performa karate memiliki waktu penyelesaian sebuah alur “Microservice Auth” dengan waktu 9.516 milisecond dan katalon menyelesaikan alur tersebut dengan waktu $15.702 \text{ millisecond}$, nilai tersebut direpresentasikan pada grafik 4.1 berikut.



Grafik 4.1 Hasil Automation Microservice Auth

Hasil automation Microservice pada Grafik 4.1 dapat diperoleh persentase waktu yang dibutuhkan oleh kedua tools sebagai berikut :

Waktu pada tools karate : 9.516 ; waktu pada tools katalon : 15.702

$$Nilai\ Efisiensi = \frac{[Nilai\ Katalon] - [Nilai\ karate]}{[Nilai\ Katalon]} \times 100\%$$

$$Nilai\ Efisiensi = \frac{15702 - 9516}{15702} \times 100\%$$

Nilai Efisiensi = 70,63%

Hal tersebut menggambarkan bahwa waktu yang dibutuhkan oleh tools karate untuk menyelesaikan alur Microservice Auth lebih rendah sebesar 70,63% dibandingkan dengan tools katalon.

2. Microservice Master

Pada Microservice Master, skenario yang dijalankan pada kedua tools automation testing sebagai berikut :

- Inquiry Uker
- Inquiry List Uker
- Inquiry List Main Uker
- Inquiry List Region Uker
- Inquiry Uker Address
- Inquiry Set Uker Address
- Inquiry Pekerjaan
- Inquiry Pekerja
- Inquiry List Pekerja
- Inquiry List Jabatan Pekerja
- Inquiry List Org Unit Pekerja
- Inquiry List Org Jabatan Pekerja
- Inquiry Pekerja OutSource
- Inquiry Pekerja List OutSource
- Inquiry Kode Pos
- Inquiry Bank
- Inquiry Hubungan Bank
- Inquiry Jenis Bank Garansi
- Inquiry Bidang Pekerjaan
- Inquiry Customer Type Badan Usaha
- Inquiry Sub Customer Type Badan Usaha
- Inquiry Data Non Finansial Mikro
- Inquiry Data Non Finansial Ritel
- Inquiry Gelar
- Inquiry Gelar Badan Usaha
- Inquiry Jenis Badan Usaha
- Inquiry Golongan Debitur
- Inquiry Golongan Debitur LBU
- Inquiry Jabatan
- Inquiry Sekon Tujuan Penggunaan Mikro
- Inquiry Sekon Tujuan Penggunaan KPR
- Inquiry Sekon Tujuan Penggunaan Ritel
- Inquiry Jenis Kredit LBU
- Inquiry Jenis Pekerjaan
- Inquiry Jenis Penggunaan LBU
- Inquiry Kategori Portfolio Badan Usaha
- Inquiry Kategori Kredit LBU
- Inquiry Mapping Loan Type
- Inquiry Mapping Loan Type New
- Inquiry Jenis Pembinaan
- Inquiry Jenis Pengikatan
- Inquiry Sifat Kredit
- Inquiry Sifat Kredit LBU
- Inquiry Produk Pinjaman
- Inquiry Status Prakarsa
- Inquiry Orientasi Penggunaan
- Inquiry Tujuan Penggunaan
- Inquiry Akua Kelas Konstruksi
- Inquiry Akua Merk Tipe Jenis Kendaraan
- Inquiry Akua Valuta
- Inquiry Akua Jenis Penggunaan
- Inquiry Akua Okupasi Alat Berat
- Inquiry Akua Warna Kendaraan
- Inquiry Akua Jenis Kendaraan
- Inquiry Penilai Independen
- Inquiry Promo Briguna
- Inquiry Objek Penggunaan Okupasi

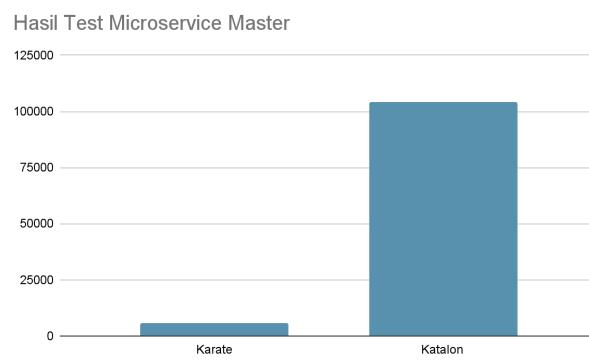
- Inquiry Perusahaan Asuransi
 - Inquiry Perusahaan Asuransi Kredit
 - Inquiry Mitra Briguna
 - Inquiry List Mitra Briguna
 - Inquiry Akua Pasar
 - Inquiry Param Biaya

- Inquiry Sektor KUR Khusus
 - Inquiry Kelompok SIKP
 - Inquiry Jenis Pengajuan Kredit
 - Inquiry Party Code
 - Inquiry JHRATE

Dari skenario tersebut didapatkan hasil yang ditampilkan pada tabel 4.1 berikut

Tabel 4.2 Hasil Test Microservice Master

Berdasarkan hasil test yang ditampilkan pada tabel 4.2 di atas diketahui bahwa performa karate memiliki waktu penyelesaian sebuah alur “Microservice Master” dengan waktu 5.927 *milisecond* dan katalon menyelesaikan alur tersebut dengan waktu 104.123 *millisecond*, nilai tersebut direpresentasikan pada grafik 4.1 berikut.



Grafik 4.2 Hasil automation Microservice Master

Hasil automation Microservice pada Grafik 4.2 dapat diperoleh persentase waktu yang dibutuhkan oleh kedua tools sebagai berikut :

Waktu pada tools karate : 5.927 ; waktu pada tools katalon : 104.123

$$Nilai Efisiensi = \frac{[Nilai Katalon] - [Nilai karate]}{[Nilai Katalon]} \times 100\%$$

$$Nilai Efisiensi = \frac{104123 - 5927}{104123} \times 100\%$$

$$Nilai Efisiensi = 94,30\%$$

Hal tersebut menggambarkan bahwa waktu yang dibutuhkan oleh tools karate untuk menyelesaikan alur Microservice Master lebih rendah sebesar 94,30% dibandingkan dengan tools katalon.

3. Microservice Eksternal

Pada Microservice Eksternal, skenario yang dijalankan pada kedua tools automation testing sebagai berikut :

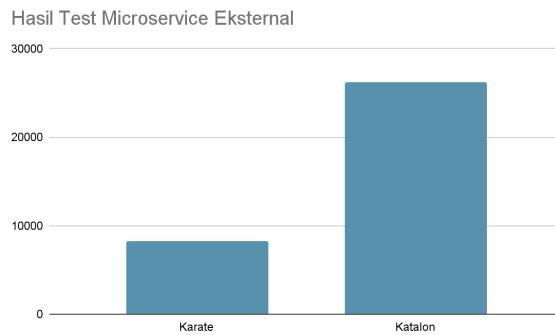
- LDAP Validate User
- Inquiry LAS User
- Notifikasi Send SMS OTP
- Notifikasi Send SMS
- Notifikasi Send SMS Email
- Login from WSAD
- Change Password from WSAD
- Login from Create User Email
- ESB Core Fund Transfer
- ESB Core Inquiry Account
- ESB Core Fund Transfer LN
- ESB Core Inquiry List Account
- ESB Core Inquiry SA
- ESB Core Inquiry CA
- ESB Core Inquiry LN
- ESB Core Inquiry TD
- ESB Core Inquiry Multi CIF Loan
- ESB Core Inquiry CIF By NIK
- ESB Core Inquiry CIF Basic Creation
- ESB Core Activation Loan
- ESB Core Loan With Drawal
- ESB Core Hold Amount
- ESB Core Unhold Amount
- ESB Core Inquiry Hold Status

Dari skenario tersebut didapatkan hasil yang ditampilkan pada tabel 4.3 berikut

Tabel 4.3 Hasil Test Microservice Eksternal

Karate	Katalon																																				
 <p>Features 2022-07-06 09:00:46 PM</p> <table border="1"> <thead> <tr> <th>Feature</th> <th>Title</th> <th>Passed</th> <th>Failed</th> <th>Scenarios</th> <th>Time (ms)</th> </tr> </thead> <tbody> <tr> <td>NeutrumD3_MicroserviceExternal 01_LDAP.feature</td> <td>03_01_Jsp</td> <td>1</td> <td>0</td> <td>1</td> <td>781</td> </tr> <tr> <td>NeutrumD3_MicroserviceExternal 02_LAS.feature</td> <td>03_01_Jsp</td> <td>1</td> <td>0</td> <td>1</td> <td>178</td> </tr> <tr> <td>NeutrumD3_MicroserviceExternal 03_NotifMail</td> <td>03_03_NotifMail</td> <td>3</td> <td>0</td> <td>3</td> <td>3082</td> </tr> <tr> <td>WSAD 03_WSAD 03_04_WSAD</td> <td>03_04_WSAD</td> <td>2</td> <td>1</td> <td>3</td> <td>1624</td> </tr> <tr> <td>ESB 03_ESB 03_06_ESB Core</td> <td>03_06_ESB Core</td> <td>8</td> <td>8</td> <td>16</td> <td>2548</td> </tr> </tbody> </table>	Feature	Title	Passed	Failed	Scenarios	Time (ms)	NeutrumD3_MicroserviceExternal 01_LDAP.feature	03_01_Jsp	1	0	1	781	NeutrumD3_MicroserviceExternal 02_LAS.feature	03_01_Jsp	1	0	1	178	NeutrumD3_MicroserviceExternal 03_NotifMail	03_03_NotifMail	3	0	3	3082	WSAD 03_WSAD 03_04_WSAD	03_04_WSAD	2	1	3	1624	ESB 03_ESB 03_06_ESB Core	03_06_ESB Core	8	8	16	2548	<p>03_Microservice_External Test Log</p> <p>Execution Environment File Lock 0% Web UI 0% API 0%</p> <p>Test Execution Log</p> <ul style="list-style-type: none"> [x] MAT-01: Microservice External [x] MAT-02: Microservice External 01_LDAP.feature [x] MAT-03: Microservice External 02_LAS.feature [x] MAT-04: Microservice External 03_NotifMail [x] MAT-05: Microservice External 04_WSAD.feature [x] MAT-06: Microservice External 05_ESB.feature [x] MAT-07: Microservice External 06_ESB Core
Feature	Title	Passed	Failed	Scenarios	Time (ms)																																
NeutrumD3_MicroserviceExternal 01_LDAP.feature	03_01_Jsp	1	0	1	781																																
NeutrumD3_MicroserviceExternal 02_LAS.feature	03_01_Jsp	1	0	1	178																																
NeutrumD3_MicroserviceExternal 03_NotifMail	03_03_NotifMail	3	0	3	3082																																
WSAD 03_WSAD 03_04_WSAD	03_04_WSAD	2	1	3	1624																																
ESB 03_ESB 03_06_ESB Core	03_06_ESB Core	8	8	16	2548																																
Karate : 8.213ms	Katalon : 26.273ms																																				

Berdasarkan hasil test yang ditampilkan pada tabel 4.3 di atas diketahui bahwa performa karate memiliki waktu penyelesaian sebuah alur “Microservice Eksternal” dengan waktu *8.213 milisecond* dan katalon menyelesaikan alur tersebut dengan waktu *26.273 millisecond*, nilai tersebut direpresentasikan pada grafik 4.3 berikut.



Grafik 4.3 Hasil automation Microservice Eksternal

Hasil automation Microservice pada Grafik 4.3 dapat diperoleh persentase waktu yang dibutuhkan oleh kedua tools sebagai berikut :

Waktu pada tools karate : 8.213 ; waktu pada tools katalon : 26.273

$$\text{Nilai Efisiensi} = \frac{[\text{Nilai Katalon}] - [\text{Nilai karate}]}{[\text{Nilai Katalon}]} \times 100\%$$

$$\text{Nilai Efisiensi} = \frac{26273 - 8213}{26273} \times 100\%$$

$$\text{Nilai Efisiensi} = 68,87\%$$

Hal tersebut menggambarkan bahwa waktu yang dibutuhkan oleh tools karate untuk menyelesaikan alur Microservice Master lebih rendah sebesar 68,87% dibandingkan dengan tools katalon.

4. Microservice Role

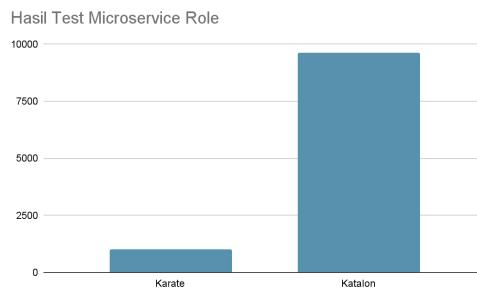
Pada Microservice role, skenario yang dijalankan pada kedua tools automation testing sebagai berikut :

- Inquiry List Menu
- Inquiry Menu
- Create Menu
- Update Menu
- Delete Menu
- Inquiry List Role
- Create Role
- Delete Role
- Inquiry List Role
- Inquiry Detail Master
- Set User to Role
- Delete User to Role
- Inquiry List Access Menu
- Role
- Set Role Menu

Dari skenario tersebut didapatkan hasil yang ditampilkan pada tabel 4.4 berikut

Tabel 4.4 Hasil Test Microservice Role

Berdasarkan hasil test yang ditampilkan pada tabel 4.4 di atas diketahui bahwa performa karate memiliki waktu penyelesaian sebuah alur “Microservice Role” dengan waktu 1.862 *milisecond* dan katalon menyelesaikan alur tersebut dengan waktu 9.638 *millisecond*, nilai tersebut direpresentasikan pada grafik 4.4 berikut.



Grafik 4.4 Hasil Automation Microservice Role

Hasil automation Microservice pada Grafik 4.4 dapat diperoleh persentase waktu yang dibutuhkan oleh kedua tools sebagai berikut :

Waktu pada tools karate : 1.862 ; waktu pada tools katalon : 9.638

$$Nilai\ Efisiensi = \frac{[Nilai\ Katalon] - [Nilai\ karate]}{[Nilai\ Katalon]} \times 100\%$$

$$Nilai\ Efisiensi = \frac{9.638 - 1862}{9.638} \times 100\%$$

Nilai Efisiensi = 80,68%

Hal tersebut menggambarkan bahwa waktu yang dibutuhkan oleh tools karate untuk menyelesaikan alur Microservice Master lebih rendah sebesar 80,68% dibandingkan dengan tools katalon.

5. Microservice Office

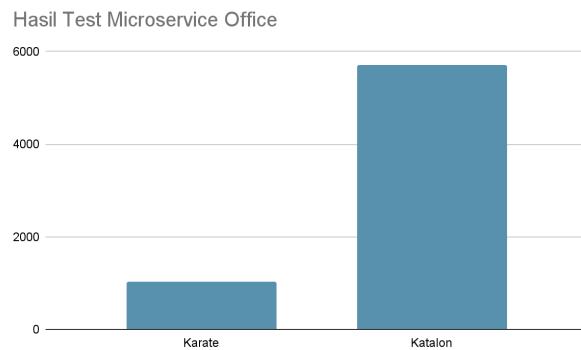
Pada Microservice Office, skenario yang dijalankan pada kedua tools automation testing sebagai berikut :

- Inquiry List Pgs
 - Inquiry Pgs
 - Inquiry Active Pgs
 - Create PGS
 - Delete PGS

Dari skenario tersebut didapatkan hasil yang ditampilkan pada tabel 4.5 berikut

Tabel 4.5 Hasil Test Microservice Office

Berdasarkan hasil test yang ditampilkan pada tabel 4.5 di atas diketahui bahwa performa karate memiliki waktu penyelesaian sebuah alur “Microservice Office” dengan waktu 1.021 *milisecond* dan katalon menyelesaikan alur tersebut dengan waktu 5.705 *milisecond*, nilai tersebut direpresentasikan pada grafik 4.5 berikut.



Grafik 4.5 Hasil automation Microservice Office

Hasil automation Microservice pada Grafik 4.5 dapat diperoleh persentase waktu yang dibutuhkan oleh kedua tools sebagai berikut :

Waktu pada tools karate : 1.021 ; waktu pada tools katalon : 5.705

$$\text{Nilai Efisiensi} = \frac{[\text{Nilai Katalon}] - [\text{Nilai karate}]}{[\text{Nilai Katalon}]} \times 100\%$$

$$\text{Nilai Efisiensi} = \frac{5705 - 1021}{5705} \times 100\%$$

$$\text{Nilai Efisiensi} = 82,10\%$$

Hal tersebut menggambarkan bahwa waktu yang dibutuhkan oleh tools karate untuk menyelesaikan alur Microservice Master lebih rendah sebesar 82,10% dibandingkan dengan tools katalon.

4.2.1 Rata-rata nilai efisiensi performa eksekusi

Berdasarkan hasil performa eksekusi, dihasilkan nilai efisiensi dari masing-masing alur pengujian, hasil tersebut dijelaskan pada tabel 4.1 dibawah ini

Tabel 4.1 nilai efisiensi hasil pengujian

No.	Alur pengujian	Nilai Efisiensi(%)
1	Microservice Auth	70,63
2	Microservice Master	94,30
3	Microservice Eksternal	68,87
4	Microservice Role	80,68
5	Microservice Office	82,10

Berdasarkan tabel hasil pengujian diatas, maka didapati hasil rata-rata dengan menggunakan persamaan sebagai berikut :

$$\bar{x} = \frac{\Sigma \text{nilai efisiensi}}{n \text{ jumlah data}}$$

$$\bar{x} = \frac{70,63 + 94,30 + 68,87 + 80,68 + 82,10}{5}$$

$$\bar{x} = \frac{396,58}{5}$$

$$\bar{x} = 79,32\%$$

dari hasil perhitungan dengan persamaan rata-rata di atas, maka nilai rata-rata efisiensi penggunaan karate dibandingkan dengan katalon adalah sebesar **79,32%**

4.3 Penggunaan Sumber Daya Tools Automation

Perbandingan penggunaan sumber daya yang digunakan saat mengoperasikan kedua tools automation API testing dilakukan dengan cara mengukur statistik aktivitas

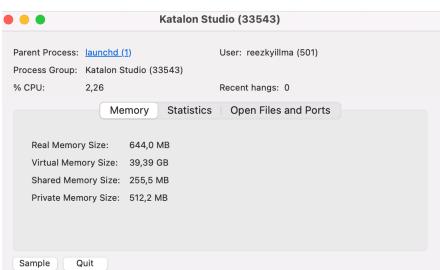
penggunaan sumber daya dari perangkat yang digunakan, pengukuran ini dilakukan pada sebuah laptop macbook dengan prosesor Apple M1 dan kapasitas RAM 8 Gigabyte.

Pengukuran dilakukan pada saat aplikasi Katalon sedang aktif dan project karate yang dibuka dengan editor IntelliJ IDEA. Pada kondisi tersebut didapat hasil sebagai berikut.

1. Penggunaan CPU

Penggunaan persentase CPU yang digunakan pada saat melakukan automation testing menggambarkan seberapa efisien sebuah tools digunakan dalam mengelola sumber daya dari potensi yang tersedia pada processor di sebuah perangkat.

Tabel 4.6 Perbandingan Penggunaan CPU

Karate	Katalon
 <p>IntelliJ IDEA (36890) Parent Process: launchd (1) User: reezkyillma (501) Process Group: IntelliJ IDEA (36890) % CPU: 1,80 Recent hangs: 0 Memory Statistics Open Files and Ports Real Memory Size: 259,2 MB Virtual Memory Size: 395,92 GB Shared Memory Size: 77,3 MB Private Memory Size: 210,5 MB Sample Quit</p>	 <p>Katalon Studio (33543) Parent Process: launchd (1) User: reezkyillma (501) Process Group: Katalon Studio (33543) % CPU: 2,26 Recent hangs: 0 Memory Statistics Open Files and Ports Real Memory Size: 644,0 MB Virtual Memory Size: 39,39 GB Shared Memory Size: 265,5 MB Private Memory Size: 512,2 MB Sample Quit</p>
Karate : 1,80 %	Katalon : 2,26%

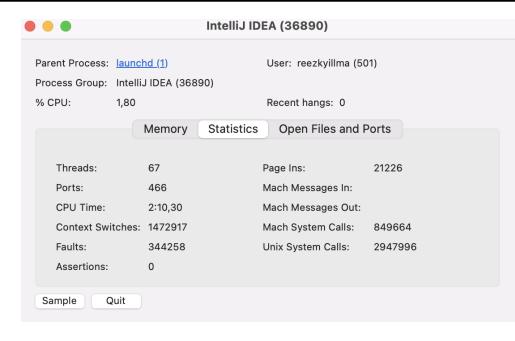
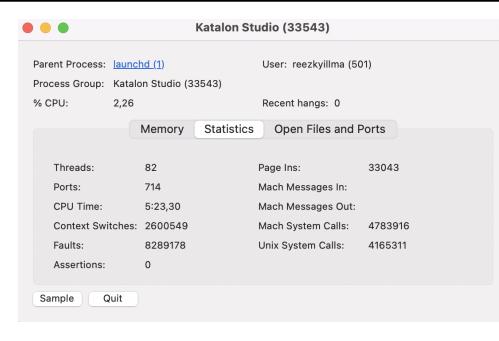
Pada tabel 4.6 diatas dijelaskan bahwa tools karate menggunakan lebih sedikit sumber daya CPU dibandingkan dengan katalon. Indikator tersebut secara langsung menunjukkan bahwa efisiensi pada penggunaan tools karate lebih baik jika dibandingkan dengan katalon.

2. Jumlah Thread Aktif

Thread merupakan unit dasar dari pemanfaatan sebuah CPU. Katalon dan IntelliJ IDEA yang menjalankan project karate adalah aplikasi yang mendukung penggunaan *multi-threading process* pada pengoperasiannya. Berdasarkan data hasil pengujian pada penggunaan persentase CPU, jumlah thread aktif memiliki korelasi yang kuat karena semakin tinggi jumlah thread aktif menunjukkan semakin besar penggunaan persentase CPU. Perbandingan jumlah thread aktif pada penggunaan tools katalon studio dan karate IntelliJ IDEA dapat dilihat pada tabel 4.7 berikut.

Tabel 4.7 Perbandingan Jumlah Thread Aktif

Karate	Katalon
--------	---------

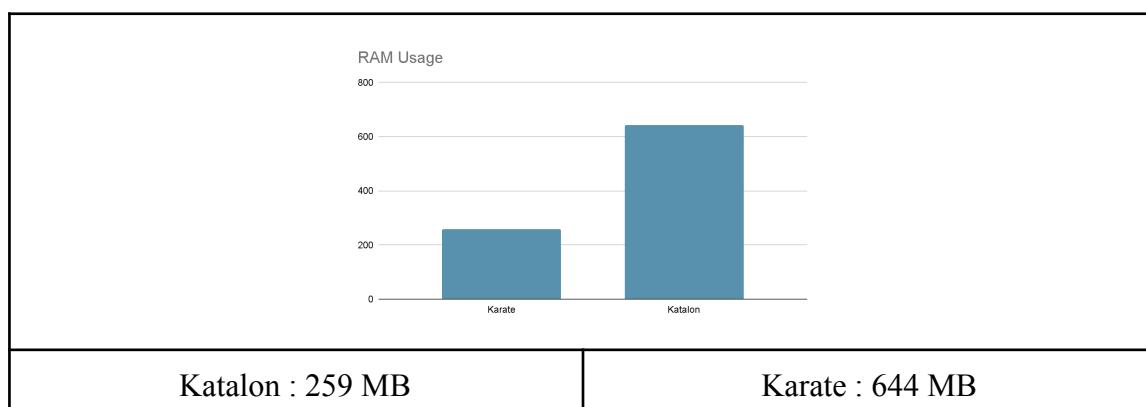
 <p>IntelliJ IDEA (36890)</p> <p>Parent Process: launchd (1) User: reezkyllima (501) Process Group: IntelliJ IDEA (36890) % CPU: 1,80 Recent hangs: 0</p> <p>Memory Statistics Open Files and Ports</p> <table border="1"> <tr><td>Threads:</td><td>67</td><td>Page Ins:</td><td>21226</td></tr> <tr><td>Ports:</td><td>466</td><td>Mach Messages In:</td><td></td></tr> <tr><td>CPU Time:</td><td>2:10,30</td><td>Mach Messages Out:</td><td></td></tr> <tr><td>Context Switches:</td><td>1472917</td><td>Mach System Calls:</td><td>849664</td></tr> <tr><td>Faults:</td><td>344258</td><td>Unix System Calls:</td><td>2947996</td></tr> <tr><td>Assertions:</td><td>0</td><td></td><td></td></tr> </table> <p>Sample Quit</p>	Threads:	67	Page Ins:	21226	Ports:	466	Mach Messages In:		CPU Time:	2:10,30	Mach Messages Out:		Context Switches:	1472917	Mach System Calls:	849664	Faults:	344258	Unix System Calls:	2947996	Assertions:	0			 <p>Katalon Studio (33543)</p> <p>Parent Process: launchd (1) User: reezkyllima (501) Process Group: Katalon Studio (33543) % CPU: 2,26 Recent hangs: 0</p> <p>Memory Statistics Open Files and Ports</p> <table border="1"> <tr><td>Threads:</td><td>82</td><td>Page Ins:</td><td>33043</td></tr> <tr><td>Ports:</td><td>714</td><td>Mach Messages In:</td><td></td></tr> <tr><td>CPU Time:</td><td>5:23,30</td><td>Mach Messages Out:</td><td></td></tr> <tr><td>Context Switches:</td><td>2600549</td><td>Mach System Calls:</td><td>4783916</td></tr> <tr><td>Faults:</td><td>8289178</td><td>Unix System Calls:</td><td>4165311</td></tr> <tr><td>Assertions:</td><td>0</td><td></td><td></td></tr> </table> <p>Sample Quit</p>	Threads:	82	Page Ins:	33043	Ports:	714	Mach Messages In:		CPU Time:	5:23,30	Mach Messages Out:		Context Switches:	2600549	Mach System Calls:	4783916	Faults:	8289178	Unix System Calls:	4165311	Assertions:	0		
Threads:	67	Page Ins:	21226																																														
Ports:	466	Mach Messages In:																																															
CPU Time:	2:10,30	Mach Messages Out:																																															
Context Switches:	1472917	Mach System Calls:	849664																																														
Faults:	344258	Unix System Calls:	2947996																																														
Assertions:	0																																																
Threads:	82	Page Ins:	33043																																														
Ports:	714	Mach Messages In:																																															
CPU Time:	5:23,30	Mach Messages Out:																																															
Context Switches:	2600549	Mach System Calls:	4783916																																														
Faults:	8289178	Unix System Calls:	4165311																																														
Assertions:	0																																																
Karate : 67	Katalon : 82																																																

Berdasarkan tabel 4.7 diketahui bahwa karate memiliki jumlah thread aktif lebih sedikit dibandingkan dengan Katalon Studio. Hal ini menunjukkan bahwa karate yang dijalankan dengan menggunakan IntelliJ IDEA memiliki nilai efisiensi penggunaan CPU lebih baik dibandingkan Katalon Studio.

3. Perbandingan penggunaan memory(RAM)

Penggunaan RAM untuk menjalankan tools automation testing sangat berpengaruh terhadap kenyamanan dan kelancaran proses pengujian. Besaran memori yang digunakan menentukan efisiensi dari sebuah tools automation testing. Perbandingan alokasi RAM antara Karate dan Katalon memiliki sifat semakin rendah penggunaan RAM maka semakin baik efisiensi dari tools tersebut. Perbandingan alokasi RAM pada tools katalon studio dan karate IntelliJ IDEA dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Perbandingan penggunaan RAM



Berdasarkan tabel 4.8 diketahui bahwa karate mengalokasikan RAM lebih sedikit dibandingkan dengan Katalon Studio. Hal ini menunjukkan bahwa karate yang dijalankan dengan menggunakan IntelliJ IDEA memiliki nilai efisiensi penggunaan RAM lebih baik dibandingkan Katalon Studio.

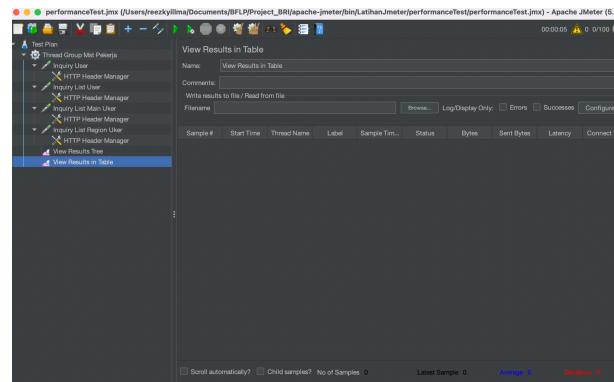
4.4 Performance Test

Performance Testing adalah proses yang menguji batas ketahanan dan kestabilan sistem serta menguji bagaimana sistem aplikasi yang digunakan dapat bekerja kembali setelah terjadi down dalam kondisi load data yang tinggi. *Performance testing* digunakan untuk mengecek apakah sistem sudah memenuhi *performance requirements*.

Performance testing yang saat ini berjalan di BRI adalah dengan menggunakan tools Apache JMeter. Tools tersebut digunakan bersamaan dengan katalon studio secara paralel dalam melakukan automation API test. Apache JMeter memiliki cara pengoperasian berbasis GUI sama seperti katalon studio. Sedangkan karate memiliki library tambahan untuk melakukan performance testing yaitu Gatling. Gatling adalah tools performance testing yang dijalankan pada framework karate. Penggunaan gatling dalam melakukan performance testing adalah dengan cara menulis script dan menjalankannya.

a. Apache JMeter

Apache JMeter adalah aplikasi open source berbasis Java yang biasa digunakan untuk performance test pada aplikasi di BRI. JMeter memiliki karakteristik pengoperasian berbasis GUI.



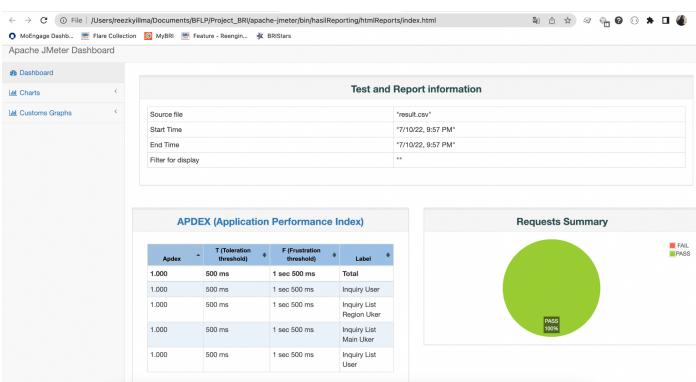
Gambar 4.1 Tampilan Aplikasi JMeter

Pada JMeter user dapat mendefinisikan test plan berdasarkan fungsi dan kategori yang akan dilakukan performance test. Struktur project yang diperlukan dalam melakukan performance test dengan menggunakan JMeter adalah sebagai berikut :

1. Test Plan berisi nama file project yang dikembangkan saat proses testing
2. Thread Group merupakan file yang mengatur konfigurasi project saat melakukan performance test
3. HTTP Request (GET, POST, PUT, DELETE) merupakan file yang berisi pengaturan http request pada API

4. HTTP Header Management adalah bagian untuk melakukan pengaturan header atau metadata yang diperlukan dalam melakukan HTTP request
5. Hasil Reporting diolah dengan menambahkan View Result Tree dan View Result Table pada thread group

Performance test ini dilakukan menggunakan API Microservice master bagian mst uker seperti check inquiry uker, inquiry list uker, inquiry list main uker, dan inquiry list region uker dengan melakukan performance test sebanyak 100 data dalam waktu 5 detik yang menampilkan hasil sebagai berikut :



Gambar 4.2 Hasil Performance Test JMeter

Pada gambar 4.2 dijelaskan mengenai informasi yang dihasilkan JMeter setelah performance test selesai dilakukan. Pada report tersebut berisi rangkuman persentase keberhasilan http request yang digambarkan dengan pie chart, waktu mulai, waktu selesai, durasi waktu http request dalam mendapatkan response, dan data lainnya yang terdapat pada tab “Chart” dan “Custom Graph”.

b. Gatling

Gatling adalah sebuah testing tools yang digunakan untuk melakukan pengujian performa sebuah API. Gatling berjalan dengan menggunakan bahasa pemrograman Scala, yaitu bahasa pemrograman yang mengimplementasi konsep *Objek Oriented Programming*.

File scala berisi karate protocol dan keterangan jumlah user dan durasi eskalasi yang akan dijalankan pada test tersebut.

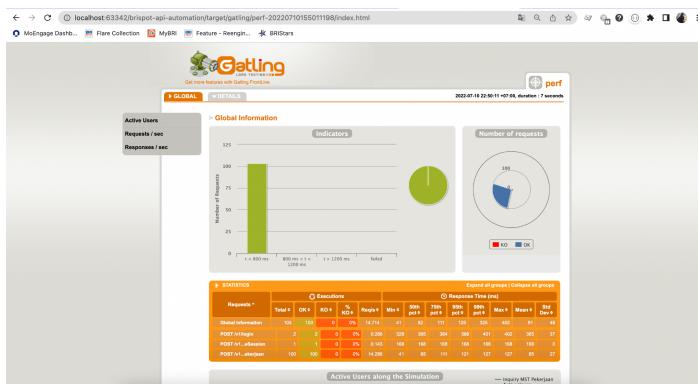
```

1 package performance
2
3 import com.intuit.karate.gatling.PreDef_
4 import io.gatling.core.Predef._
5 import scala.concurrent.duration._
6
7 class Perf extends Simulation {
8
9   val protocol = karateProtocol()
10
11   val InquiryMSTPekerjaan = scenario(scenarioName = "Inquiry MST Pekerjaan")
12     .exec(karateFeature(name = "classpath:features/02_MicroserviceMaster/03_mst_pekerjaan.feature"))
13
14   setUp(
15     inquiryMSTPekerjaan.inject(rampUsers(users = 100) during(5 seconds)).protocols(protocol)
16   )
17 }

```

Gambar 4.3 File Scala pada Performance Test Karate

Gambar 4.3 menjelaskan struktur file yang ada di dalam file scala. Pada gambar tersebut menampilkan bahwa untuk membuat sebuah skenario performance test pengguna diperlukan mengimport library yang digunakan, mendefinisikan class yang mengimplementasi method dari library “Simulation”, Tampilan script Gatling menunjukkan bahwa untuk melakukan performance test, pengguna terlebih dahulu membuat skenario http request, mendefinisikan berapa user yang disimulasikan serta durasi eskalasi waktu yang diperlukan, selanjutnya request tersebut dipanggil pada runner dengan format file *.scala. Performance test ini dilakukan menggunakan API Microservice master bagian mst uker seperti check inquiry uker, inquiry list uker, inquiry list main uker, dan inquiry list region uker dengan melakukan performance tense sebanyak 100 data dalam waktu 5 detik yang menampilkan hasil sebagai berikut:



Gambar 4.4 Hasil Performance Test Karate-Gatling

Pada gambar 4.4 menjelaskan mengenai hasil dari performance test menggunakan karate gatling secara global yang berisi informasi rangkuman persentase keberhasilan http request yang digambarkan dengan line chart, pie chart, waktu mulai, waktu selesai, durasi waktu http request dalam mendapatkan response, dan data lainnya secara ringkas mengenai detail informasi request.

Secara garis besar dapat disimpulkan bahwa perbedaan penggunaan tools Gatling dan Jmeter tidak memiliki perbedaan yang signifikan dari segi performance. Tools Jmeter menggunakan CPU lebih tinggi dibandingkan dengan Gatling dikarenakan Jmeter berbasis GUI sedangkan gatling berbasis command line runner. Tools JMeter lebih sulit diintegrasikan dengan aplikasi lain dibandingkan dengan Gatling karena tools JMeter yang bersifat independen. Proses penggunaan aplikasi JMeter berbasis GUI memiliki tampilan yang lebih baik dibandingkan dengan Gatling.

c. Rata-rata Nilai Efisiensi Performance Test

Berdasarkan hasil performance test penggunaan tools JMeter dan karate, dihasilkan nilai efisiensi dari masing-masing alur pengujian, hasil tersebut dijelaskan pada tabel 4.1 dibawah ini

d. Tabel 4.1 nilai efisiensi hasil pengujian

No.	Alur pengujian	Waktu Pengujian (s)
1	Request 100 User dengan JMeter	15 second
2	Request 100 User dengan Karate	6 second

Berdasarkan tabel hasil pengujian diatas, maka didapati hasil rata-rata dengan menggunakan persamaan sebagai berikut :

$$\text{Nilai Efisiensi Tools Karate} = \frac{\Sigma \text{nilai efisiensi JMeter} - \Sigma \text{nilai efisiensi Karate}}{\Sigma \text{nilai efisiensi JMeter}}$$

$$\text{Nilai Efisiensi Tools Karate} = \frac{7}{15}$$

$$\text{Nilai Efisiensi Tools Karate} = 60\%$$

Berdasarkan hasil perhitungan nilai efisiensi di atas, maka nilai efisiensi penggunaan karate dibandingkan dengan Jmeter sebesar **60%**. Pada proses automation testing, aspek integrasi dan *portability* memiliki pengaruh yang signifikan, karena produktivitas seorang automation tester dapat terganggu jika harus berpindah konteks pekerjaan, sebaliknya jika automation tester bekerja pada satu ekosistem yang sama dapat meningkatkan produktivitas yang baik. Hal ini berkaitan erat dengan konsep *single environment* yang digunakan pada tools karate. Konsep tersebut memungkinkan seorang automation tester dapat bekerja dengan satu ekosistem terpadu. Contoh kasus yang dapat diberikan adalah saat automation tester telah selesai membuat skenario untuk pengujian API, maka skenario tersebut dapat digunakan

kembali pada performance test dengan menggunakan gatling tanpa perlu merancang ulang skenario untuk proses performance test.

4.5 Usability

Faktor kegunaan dari tools automation API testing ditentukan dari seberapa baik tingkat usability tools tersebut pada kondisi yang sebenarnya. Kondisi yang dimaksud mencakup beberapa faktor diantaranya :

a. Kemudahan Mempelajari Tools

Proses testing memiliki tujuan untuk mengetahui seberapa efektif program atau aplikasi yang dikembangkan sebelum dirilis ke publik. Pada proses testing diperlukan tools automation testing dalam meningkatkan produktivitas testing dan meminimalisir bugs pada aplikasi, sehingga mendapatkan hasil testing yang akurat. Kemudahan mempelajari tools memiliki pengaruh kepada pengguna dalam merancang alur dari proses testing yang dikembangkan.

Pada katalon studio memiliki kelebihan yaitu sebagai tools yang berlisensi free dan user friendly. Tools katalon studio memiliki fitur merekam langkah pengujian, menangkap elemen, dan memberikan hasil laporan otomatis dalam format html, pdf, dan csv pada langkah pengujian yang sudah didefinisikan di aplikasi. Katalon studio juga dapat melakukan proses testing Web, Mobile dan API. Kelemahan tools katalon studio terletak pada penggunaan script groovy yang bersifat closed source code, penggunaan memory yang cukup tinggi, dan dokumentasi katalon studio yang tidak dijelaskan secara detail.

Dibandingkan dengan katalon studio, karate memiliki kelebihan sebagai tools yang berlisensi open source dengan mengkombinasikan proses API automation, mocks testing, performance-testing and web UI testing dalam satu framework, Pengujian pada tools karate menggunakan script BDD Cucumber seperti natural language yang dapat memudahkan non programmer untuk memahaminya, test report yang dihasilkan berupa sebuah web page berformat html sehingga lebih interaktif, dan script yang dijalankan dapat dilakukan secara paralel. Dari kelebihan karate yang telah dijabarkan diatas, karate juga memiliki kelemahan diantaranya tidak memiliki tampilan user friendly secara GUI, karate memerlukan text editor tambahan untuk merancang script testing, dan tidak dapat melakukan mobile automation testing seperti tools katalon.

b. Cakupan Objek Testing

Salah satu faktor yang menjadi pertimbangan untuk menggunakan tools automation testing adalah cakupan objek testing yang terdapat pada tools tersebut. Katalon studio memiliki cakupan objek testing yang lebih luas, hal itu dapat dibuktikan dengan kemampuan katalon studio dalam melakukan pengujian Web app, Mobile app, dan API dalam satu aplikasi. Fleksibilitas yang dimiliki katalon studio tersebut tentu memiliki kelemahan yang telah dijelaskan pada penjelasan sebelumnya mengenai tingginya persentase penggunaan CPU dan memori(RAM).

Karate memiliki kelebihan dalam efisiensi penggunaan sumberdaya dan performanya dalam melakukan pengujian API dan performance test. Kelemahan yang terdapat pada karate adalah cakupan objek testing yang terbatas pada pengujian API

c. Fleksibilitas Penggunaan

Faktor fleksibilitas dapat dijadikan parameter dalam penggunaan automation API test. Fleksibilitas yang dimiliki oleh kedua tools automation API test memiliki perbedaan. Perancangan test case pada katalon studio dilakukan dengan menggunakan aplikasi katalon studio itu sendiri, sehingga pengguna kurang memiliki keleluasaan untuk menentukan editor.

Karate dijalankan dengan teks editor untuk merancang test case, terdapat beberapa pilihan teks editor yang dapat digunakan untuk merancang test case contohnya adalah IntelliJ IDEA, Visual Studio Code, dan lain-lain.

d. Detail informasi pada Report hasil testing

Salah satu faktor lainnya seperti detail informasi yang dihasilkan dari proses pengujian menjadi pertimbangan dalam mengimplementasikan tools automation testing. Pada katalon studio memiliki format reporting berbentuk file PDF, HTML, dan CSV. Bentuk reporting tersebut menjelaskan tahapan proses testing yang telah didefinisikan pada test case. Sedangkan reporting pada karate hanya memiliki format reporting file HTML dengan detail informasi dan respon yang didefinisikan pada file feature. File HTML yang dihasilkan oleh karate sebagai output reporting memiliki informasi yang lebih detail mengenai langkah-langkah yang dilakukan oleh karate dan informasi tersebut disampaikan dengan susunan BDD cucumber, sehingga laporan menjadi lebih informatif untuk pengguna.

e. Detail informasi Lisensi Tools

Lisensi Software adalah hak eksklusif pembuat atau pemilik software atas izin, hak dan pembatasan untuk perangkat lunaknya. Sehingga software dapat digunakan, disebarluaskan, diperbanyak atau diubah oleh pihak lain dengan berpedoman pada

peraturan yang tercantum pada lisensi software tersebut. Penggunaan lisensi yang terdapat pada katalon dan karate sangat penting diperhatikan dalam mendukung pekerjaan. Tools Karate memiliki lisensi berjenis *open source* sehingga pengguna dapat menggunakan fitur tersebut secara bebas. Sedangkan pada tools katalon studio memiliki beberapa lisensi sebagai berikut :

1. Katalon Studio bersifat Open Source
 - Web testing
 - API testing
 - Mobile testing
 - Desktop testing
2. Katalon Studio Enterprise dengan biaya \$1,899
 - All Studio functions
 - Extended features
 - Private plugins
 - Helpdesk support
3. Katalon Runtime Engine
 - Enable CI/CD integration
 - Support all execution environments
 - Schedule & run tests in parallel

Berdasarkan data tersebut, membuktikan bahwa karate memiliki kemudahan dalam mengakses keseluruhan fitur yang ada dibandingkan dengan katalon studio yang memiliki akses terbatas pada lisensi gratis.

4.5.1 Hasil Kuesioner

Penyusunan kuesioner adalah langkah yang dilakukan untuk mengetahui kepuasan satu pengguna yang dinilai berdasarkan karakteristik evaluasi yang ditentukan dalam kuesioner tersebut. Penyusunan pertanyaan di dalam kuesioner dengan menetapkan informasi dari pengukuran usability yang sudah ditentukan sebelumnya. Perancangan kuesioner memiliki beberapa karakteristik yang damat menjadi sataşan permasalahan dalam penelitian. Berikut tahapan singkat dalam penyusunan kuesioner sebagai berikut :

- | | |
|---------------------------------------|---|
| 1. Menetapkan Informasi | 5. Menentukan Target Responden |
| 2. Menentukan Jenis Kuesioner | 6. Menyusun Kuesioner |
| 3. Menentukan Target Kuesioner | 7. Menyebar Kuesioner |
| 4. Menentukan Karakteristik Responden | 8. Mengumpulkan Hasil dari Kuesioner Penelitian |

Penyebaran kuesioner dilakukan kepada beberapa user yang pernah menggunakan Automation API Testing pada tools Katalon dan tools Karate yang dapat memberikan jawaban terkait usability dari tools Automation Testing tersebut. Proses penyebaran kuesioner melalui link google form yang memiliki beberapa pertanyaan terkait penelitian analisis perbandingan tools Karate dan tools Katalon Studio dalam melakukan API Automation. Kuesioner ini, memiliki tiga bagian dengan menggunakan tiga jenis form jawaban yang berbeda antara halaman 1 dengan halaman lainnya.

1. Pada bagian halaman 1 Mengenai karakteristik dari responden.
2. Halaman 2 mengenai cara mengisi pertanyaan yang ditentukan dengan beberapa point seperti :

Tabel 4.9 Bobot Penilaian Jawaban Kuesioner

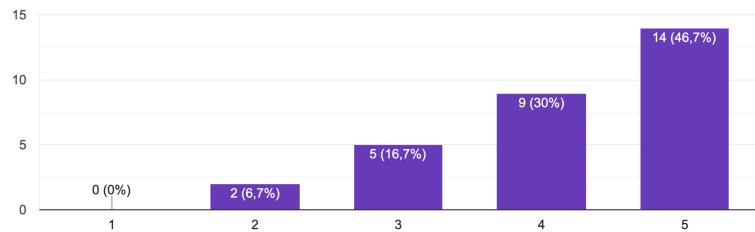
Jawaban Penilaian	Bobot Penilaian
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu-Ragu	3
Setuju	4
Sangat Setuju	5

3. Halaman 3 mengenai pertanyaan terkait tolak ukur perbandingan usability pada tools karate dan tools Katalon Studio dengan jawaban menggunakan skala linier pada setiap pertanyaan.

Kuesioner tersebut, menerapkan pertanyaan yang wajib untuk dijawab dengan tanda bintang (*) pada setiap pertanyaan dan memiliki hasil kuesioner dari 30 responden. Berdasarkan data hasil kuesioner, maka data tersebut diolah dengan mengalikan setiap poin jawaban dengan bobot yang telah ditentukan, sehingga hasil perhitungan untuk pertanyaan sebagai berikut :

1. Hasil Kuesioner Pertanyaan ke-1

Apakah tools karate mudah di pelajari dibandingkan dengan tools Katalon Studio?
30 jawaban



Gambar 4.5 Hasil dari Kuesioner Pertanyaan ke-1

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 14 \times 5$$

$$\text{Total Responden Sangat Setuju} = 70$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 9 \times 4$$

$$\text{Total Responden Setuju} = 36$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 10 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 30$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 2 \times 2$$

$$\text{Total Responden Kurang Setuju} = 4$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

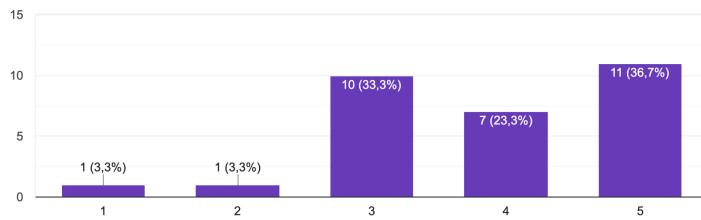
Hasil kuesioner dari pertanyaan ke 1 sebagai berikut :

$$\text{Hasil Kuesioner ke - 1} = 70 + 36 + 15 + 4 + 0$$

$$\text{Hasil Kuesioner ke - 1} = 125$$

2. Hasil Kuesioner Pertanyaan ke-2

Apakah tools karate memiliki cakupan objek testing lebih luas dibandingkan dengan tools Katalon Studio?
30 jawaban



Gambar 4.6 Hasil dari Kuesioner Pertanyaan ke-2

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 11 \times 5$$

$$\text{Total Responden Sangat Setuju} = 55$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 7 \times 4$$

$$\text{Total Responden Setuju} = 28$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 10 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 15$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 1 \times 2$$

$$\text{Total Responden Kurang Setuju} = 2$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 1 \times 1$$

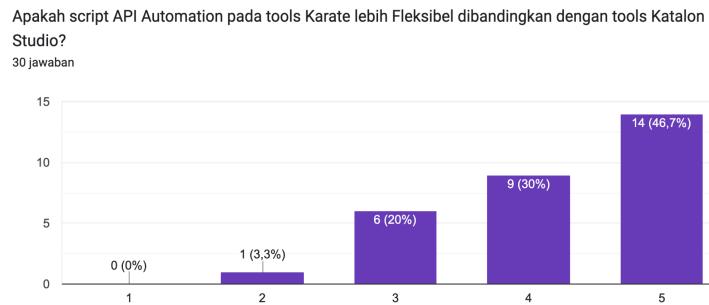
$$\text{Total Responden Tidak Setuju} = 1$$

Hasil kuesioner dari pertanyaan ke 2 sebagai berikut :

$$\text{Hasil Kuesioner ke - 2} = 55 + 28 + 30 + 2 + 1$$

$$\text{Hasil Kuesioner ke - 2} = 116$$

3. Hasil Kuesioner Pertanyaan ke-3



Gambar 4.7 Hasil dari Kuesioner Pertanyaan ke-3

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 14 \times 5$$

$$\text{Total Responden Sangat Setuju} = 70$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 9 \times 4$$

$$\text{Total Responden Setuju} = 36$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 6 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 18$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 1 \times 2$$

$$\text{Total Responden Kurang Setuju} = 2$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

Hasil kuesioner dari pertanyaan ke 3 sebagai berikut :

$$\text{Hasil Kuesioner ke - 3} = 70 + 36 + 18 + 2 + 0$$

$$\text{Hasil Kuesioner ke - 3} = 126$$

4. Hasil Kuesioner Pertanyaan ke-4



Gambar 4.8 Hasil dari Kuesioner Pertanyaan ke-4

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 16 \times 5$$

$$\text{Total Responden Sangat Setuju} = 80$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 6 \times 4$$

$$\text{Total Responden Setuju} = 24$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 7 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 21$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 1 \times 2$$

$$\text{Total Responden Kurang Setuju} = 2$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

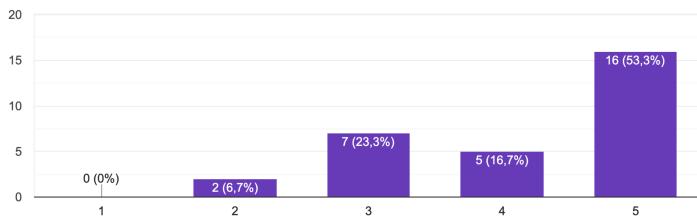
Hasil kuesioner dari pertanyaan ke 4 sebagai berikut :

$$\text{Hasil Kuesioner ke - 4} = 80 + 24 + 21 + 2 + 0$$

$$\text{Hasil Kuesioner ke - 4} = 127$$

5. Hasil Kuesioner Pertanyaan ke-5

Apakah struktur project dari tools Karate lebih mudah di mengerti dibandingkan dengan tools Katalon Studio?
30 jawaban



Gambar 4.9 Hasil dari Kuesioner Pertanyaan ke-5

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 16 \times 5$$

$$\text{Total Responden Sangat Setuju} = 80$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 5 \times 4$$

$$\text{Total Responden Setuju} = 20$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 7 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 21$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 2 \times 2$$

$$\text{Total Responden Kurang Setuju} = 4$$

5. Jumlah responden yang menjawab Tidak Setuju

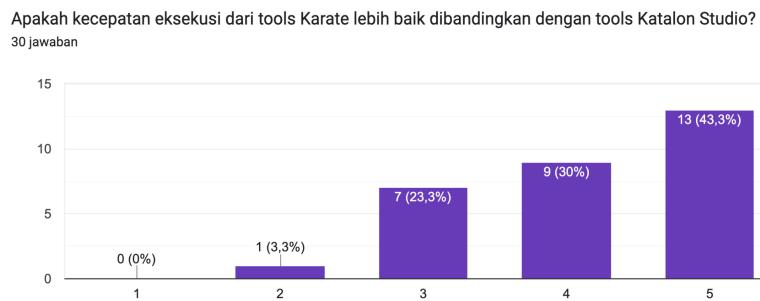
$$\text{Total Responden Tidak Setuju} = 0$$

Hasil kuesioner dari pertanyaan ke 5 sebagai berikut :

$$\text{Hasil Kuesioner ke - 5} = 80 + 20 + 21 + 4 + 0$$

$$\text{Hasil Kuesioner ke - 5} = 125$$

6. Hasil Kuesioner Pertanyaan ke-6



Gambar 4.10 Hasil dari Kuesioner Pertanyaan ke-6

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 13 \times 5$$

$$\text{Total Responden Sangat Setuju} = 65$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 9 \times 4$$

$$\text{Total Responden Setuju} = 36$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 7 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 21$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 1 \times 2$$

$$\text{Total Responden Kurang Setuju} = 2$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

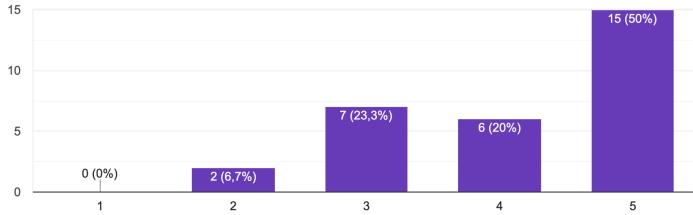
Hasil kuesioner dari pertanyaan ke 6 sebagai berikut :

$$\text{Hasil Kuesioner ke - 6} = 65 + 36 + 21 + 2 + 0$$

$$\text{Hasil Kuesioner ke - 6} = 124$$

7. Hasil Kuesioner Pertanyaan ke-7

Apakah hasil reporting dari tools Karate lebih detail dan mudah dipahami dibandingkan dengan Katalon Studio?
30 jawaban



Gambar 4.11 Hasil dari Kuesioner Pertanyaan ke-6

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 15 \times 5$$

$$\text{Total Responden Sangat Setuju} = 75$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 6 \times 4$$

$$\text{Total Responden Setuju} = 24$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 7 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 21$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 2 \times 2$$

$$\text{Total Responden Kurang Setuju} = 4$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

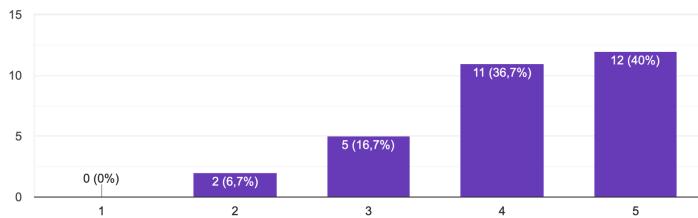
Hasil kuesioner dari pertanyaan ke 7 sebagai berikut :

$$\text{Hasil Kuesioner ke - 7} = 75 + 24 + 21 + 4 + 0$$

$$\text{Hasil Kuesioner ke - 7} = 124$$

8. Hasil Kuesioner Pertanyaan ke-8

Apakah dalam melakukan performance testing menggunakan tools Karate lebih mudah dibandingkan dengan tools eksisting BRI seperti JMeter?
30 jawaban



Gambar 4.12 Hasil dari Kuesioner Pertanyaan ke-8

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 12 \times 5$$

$$\text{Total Responden Sangat Setuju} = 60$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 11 \times 4$$

$$\text{Total Responden Setuju} = 44$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 5 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 15$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 2 \times 2$$

$$\text{Total Responden Kurang Setuju} = 4$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

Hasil kuesioner dari pertanyaan ke 8 sebagai berikut :

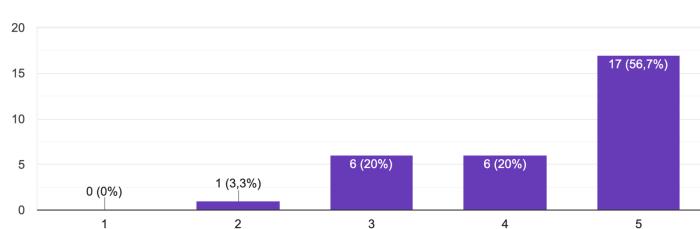
$$\text{Hasil Kuesioner ke - 8} = 60 + 44 + 15 + 4 + 0$$

$$\text{Hasil Kuesioner ke - 8} = 123$$

9. Hasil Kuesioner Pertanyaan ke-9

Apakah script untuk melakukan performance testing pada tools Karate lebih mudah dipahami dibandingkan dengan tools JMeter?

30 jawaban



Gambar 4.13 Hasil dari Kuesioner Pertanyaan ke-9

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 17 \times 5$$

$$\text{Total Responden Sangat Setuju} = 85$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 6 \times 4$$

$$\text{Total Responden Setuju} = 24$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 6 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 18$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 1 \times 2$$

$$\text{Total Responden Kurang Setuju} = 2$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

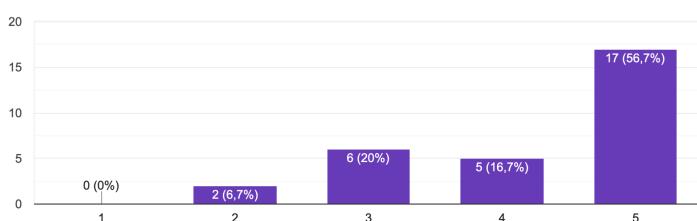
Hasil kuesioner dari pertanyaan ke 9 sebagai berikut :

$$\text{Hasil Kuesioner ke - 9} = 85 + 24 + 18 + 2 + 0$$

$$\text{Hasil Kuesioner ke - 9} = 129$$

10. Hasil Kuesioner Pertanyaan ke-10

Apakah hasil report dari performance testing menggunakan tools Karate lebih detail dan lebih dimengerti dibandingkan dengan tools JMeter?
30 jawaban



Gambar 4.14 Hasil dari Kuesioner Pertanyaan ke-10

1. Jumlah responden yang menjawab Sangat Setuju

$$\text{Total Responden Sangat Setuju} = 17 \times 5$$

$$\text{Total Responden Sangat Setuju} = 85$$

2. Jumlah responden yang menjawab Setuju

$$\text{Total Responden Setuju} = 5 \times 4$$

$$\text{Total Responden Setuju} = 20$$

3. Jumlah responden yang menjawab Ragu-ragu

$$\text{Total Responden Ragu - Ragu} = 6 \times 3$$

$$\text{Total Responden Ragu - Ragu} = 18$$

4. Jumlah responden yang menjawab Kurang Setuju

$$\text{Total Responden Kurang Setuju} = 2 \times 2$$

$$\text{Total Responden Kurang Setuju} = 4$$

5. Jumlah responden yang menjawab Tidak Setuju

$$\text{Total Responden Tidak Setuju} = 0$$

Hasil kuesioner dari pertanyaan ke 10 sebagai berikut :

$$\text{Hasil Kuesioner ke - 10} = 85 + 20 + 18 + 4 + 0$$

$$\text{Hasil Kuesioner ke - 10} = 127$$

Hasil interpretasi dari kuesioner analisis perbandingan tools karate dan katalon dilakukan dengan cara mengetahui skor tertinggi Y dan angka terendah X untuk setiap penilaian menggunakan rumus sebagai berikut :

1. Perhitungan dengan skor tertinggi

$$Y = SkorLikertTertinggi * JumlahResponden$$

$$SkorLikertTertinggi = 5$$

$$Y = 5 * 30$$

$$Y = 150$$

2. Perhitungan dengan skor terendah

$$X = SkorLikertTerendah * JumlahResponden$$

$$SkorLikertTerendah = 1$$

$$X = 1 * 30$$

$$X = 30$$

Jumlah skor tertinggi untuk pilihan sangat setuju adalah 150, sedangkan jumlah skor pilihan tidak setuju adalah 30. Dari hasil tersebut, perhitungan untuk interpretasi indeks sebagai berikut :

$$Indeks = \frac{TotalSkor}{Y} \times 100\%$$

1. Total Responden pertanyaan ke-1

$$Indeks = 84\%$$

$$Hasil Kuesioner ke - 1 = 125$$

$$Indeks = \frac{125}{150} \times 100\%$$

$$Indeks = 83.33\%$$

4. Total Responden pertanyaan ke-4

$$Hasil Kuesioner ke - 4 = 127$$

$$Indeks = \frac{127}{150} \times 100\%$$

2. Total Responden pertanyaan ke-2

$$Indeks = 84.66\%$$

$$Hasil Kuesioner ke - 2 = 116$$

$$Indeks = \frac{116}{150} \times 100\%$$

$$Indeks = 77.33\%$$

5. Total Responden pertanyaan ke-5

$$Hasil Kuesioner ke - 5 = 125$$

$$Indeks = \frac{125}{150} \times 100\%$$

3. Total Responden pertanyaan ke-3

$$Indeks = 83.33\%$$

$$Hasil Kuesioner ke - 3 = 126$$

$$Indeks = \frac{126}{150} \times 100\%$$

6. Total Responden pertanyaan ke-6

$$Hasil Kuesioner ke - 6 = 124$$

$$Indeks = \frac{124}{150} \times 100\%$$

$$Indeks = 83.33\%$$

$$Indeks = 82.66\%$$

9. Total Responden pertanyaan ke-9

$$Hasil Kuesioner ke - 9 = 129$$

7. Total Responden pertanyaan ke-7

$$Hasil Kuesioner ke - 7 = 124$$

$$Indeks = \frac{129}{150} \times 100\%$$

$$Indeks = \frac{124}{150} \times 100\%$$

$$Indeks = 82\%$$

$$Indeks = 82.66\%$$

10. Total Responden pertanyaan ke-10

$$Hasil Kuesioner ke - 10 = 127$$

8. Total Responden pertanyaan ke-8

$$Hasil Kuesioner ke - 8 = 123$$

$$Indeks = \frac{127}{150} \times 100\%$$

$$Indeks = \frac{123}{150} \times 100\%$$

$$Indeks = 84.66\%$$

Hasil perhitungan dari data kuesioner sebagai berikut :

Tabel 4.10 Hasil Data Kuesioner

No	Pertanyaan	STS (1)	TS (2)	RR (3)	S (4)	SS (5)	Total	Indeks
1.	Apakah tools karate mudah dipelajari dibandingkan dengan tools Katalon Studio?	0	2	5	9	14	125	83.33%
2.	Apakah tools karate memiliki cakupan objek testing lebih luas dibandingkan dengan tools Katalon Studio?	1	2	10	7	11	116	77.33%
3.	Apakah script API Automation pada tools Karate lebih Fleksibel dibandingkan dengan tools Katalon Studio?	0	1	6	9	14	126	84%
4.	Apakah penggunaan sumber daya pada tools Karate lebih efektif dibandingkan dengan tools Katalon Studio?	0	1	7	6	16	127	84.66%
5.	Apakah struktur project dari tools Karate lebih mudah dimengerti	0	2	7	5	16	125	83.33%

	dibandingkan dengan tools Katalon Studio?						
6.	Apakah kecepatan eksekusi dari tools Karate lebih baik dibandingkan dengan tools Katalon Studio?	0	1	7	9	13	124 82.66%
7.	Apakah hasil reporting dari tools Karate lebih detail dan mudah dipahami dibandingkan dengan Katalon Studio?	0	2	7	6	15	124 82.66%
8.	Apakah dalam melakukan performance testing menggunakan tools Karate lebih mudah dibandingkan dengan tools eksisting BRI seperti JMeter?	0	2	5	11	12	123 83.33%
9.	Apakah script untuk melakukan performance testing pada tools Karate lebih mudah dipahami dibandingkan dengan tools JMeter?	0	1	6	6	17	129 82%
10.	Apakah hasil report dari performance testing menggunakan tools Karate lebih detail dan lebih dimengerti dibandingkan dengan tools JMeter?	0	2	6	5	17	127 84,66%
Total		1	15	68	73	145	1246 82.80%

Perhitungan Interval Skor Hasil Kuesioner sebagai berikut:

$$I = \frac{100}{Jumlah\ Likert} \times 100\%$$

$$I = \frac{100}{5} \times 100\%$$

$$I = 20\%$$

- Angka 80%-100% : Sangat Setuju
- Angka 60%-80% : Setuju
- Angka 40%-60% ; Ragu-Ragu

- Angka 20%-40% : Kurang Setuju
- Angka 0%-20% : Tidak Setuju

Penyelesaian akhir menggunakan rumus Indeks dengan skala keseluruhan responden sebagai berikut :

$$Y_{akhir} = Total Likert Tertinggi * Jumlah Pernyataan$$

$$Y_{akhir} = 150 * 10$$

$$Y_{akhir} = 1500$$

$$Indeks_{akhir} = \frac{Total Skor}{Y} \times 100\%$$

$$Indeks_{akhir} = \frac{1246}{1500} \times 100\%$$

$$Indeks_{akhir} = 83\%$$

Berdasarkan tabel 4.10 setelah melakukan kuesioner terhadap 30 responden pengguna API Automation menggunakan tools Karate dan tools Katalon Studio, maka diperoleh hasil persentase 83% yang memiliki kategori "Sangat Setuju" bahwa tools Karate lebih baik dibandingkan dengan tools Katalon Studio dalam melakukan API Automation Testing.

4.7 Hasil Pengujian

Hasil pengujian yang telah dilakukan pada perbandingan dua tools automation API dijelaskan pada tabel 4.11 berikut

Tabel 4.11 Hasil Pengujian pada Katalon Studio dan Karate

No	Jenis Pengujian	Hasil Pengujian
1	Performa	Karate memiliki performa lebih baik pada sektor kecepatan eksekusi dan durasi pengujian
2	Penggunaan Sumberdaya	Karate menggunakan sumber daya yang lebih efisien pada sektor persentase penggunaan CPU, thread aktif, dan

		penggunaan memori(RAM)
3	Struktur Project	Struktur project pada karate memiliki fleksibilitas yang tinggi dan memiliki kustomisasi lebih luas karena perancangan test script berbasis teks, pada katalon studio struktur project lebih baku dan teratur karena pengoperasian aplikasi berbasis GUI
4	Performance Test	JMeter memiliki informasi output yang lebih kompleks dan data yang lebih detail. Gatling menampilkan informasi bersifat kesimpulan yang memiliki tampilan lebih ringkas dan mudah dimengerti
5	Usability	Karate lebih mudah diimplementasikan oleh pengguna non-programmer karena menggunakan bahasa yang lebih mudah dimengerti dan alur perancangan yang ringkas. Katalon studio lebih sesuai pada pengguna yang memiliki latar belakang programming dalam hal pemahaman alur API automation.

Tabel 4.11 menjelaskan perbandingan pada masing-masing parameter yang menjadi kunci pembahasan pada penelitian ini. Dari masing-masing parameter tersebut membuktikan bahwa dalam hal proses API automation testing tools karate memiliki performa yang lebih baik dibandingkan dengan tools katalon studio, seperti yang dibuktikan pada tabel 4.11 poin nomor 1 dan 2 karate lebih unggul dari segi performa dan penggunaan sumberdaya. Dua faktor tersebut memiliki pengaruh yang signifikan jika diterapkan pada project yang sudah berjalan serta efisiensi waktu yang dibutuhkan seorang automation tester dalam melakukan tugasnya untuk menguji kualitas dari API yang dikembangkan pada sebuah project.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengamatan dan penelitian yang telah dijelaskan pada bab sebelumnya, maka dapat disimpulkan bahwa dari 5 parameter utama penelitian yaitu performa aplikasi, penggunaan sumber daya, struktur project, performance testing, dan usability, karate menunjukkan hasil positif dan melampaui hasil penilaian katalon studio, sehingga karate dapat dijadikan tools alternatif untuk melakukan automation API testing di Bank BRI.

5.2 Saran

Proses automation testing di Bank BRI dilakukan secara terbatas dengan menggunakan tools Katalon Studio dengan lisensi gratis(*Free License*). Diharapkan pada kesempatan yang akan datang, Bank BRI dapat menggunakan aplikasi lain sebagai alternatif pilihan untuk proses Automation Testing dengan fitur yang lebih baik dari tools tersebut. Salah satunya adalah penggunaan tools Karate yang dapat dijadikan rekomendasi untuk melakukan proses API Automation dan performance testing.

DAFTAR PUSTAKA

Thomas, Peter. (2019). Karate Automation Testing. Diakses pada 04 Juli 2022 dari
<https://karatelabs.github.io/karate/>

IBM. (2021). Software testing. Diakses pada 04 Juli 2022, dari
<https://www.ibm.com/topics/software-testing>

Giraldo, Santiago Cardona (2020).Karate API Testing. Diakses pada 05 Juli 2022 dari
<https://medium.com/globant/karate-api-testing-9843bbe8abe>

Bhatt, Rishil . (2017). Understand Summary Report JMeter. Diakses pada 05 Juli 2022 dari
<http://www.testingjournals.com/understand-summary-report-jmeter/>

Lumbantobing, Hariyat (2021). Analisis Hasil Performance Test pada JMeter. Diakses pada
05 Juli 2022 dari
<https://hariyatlumbantobing.medium.com/analisis-hasil-performance-test-pada-jmeter-1d7a432906d7>

Momin, Mohammad Hassaan. (2022).Karate and BDD: Fast, Open-Source API Test
Automation Done Right. Diakses pada 05 Juli 2022 dari
<https://www.mobilelive.ca/blog/karate-bdd-and-api-automation>

Ludwikowski, Andrzej. (2022). Performance Testing. Diakses pada 05 Juli 2022 dari
<https://www.infoq.com/articles/performance-testing/>

Loisel, Jerome. (2015). JMeter VS Gatling. Diakses pada 05 Juli 2022 dari
<https://octoperf.com/blog/2015/06/08/jmeter-vs-gatling/#jmeter-test-execution>